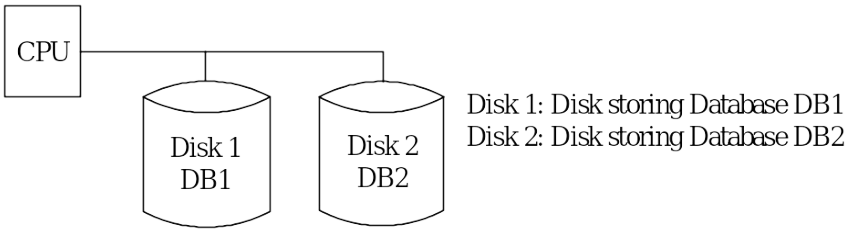
5 Công nghệ xử lý thông tin

[Phạm vi các câu hỏi]
Hiệu năng hệ thống, Tính tin cậy,
Quản lý rủi ro, Tính an toàn, Chuẩn hóa,
Các hoạt động nghiên cứu...

Câu hỏi 1

Q1. Đọc mô tả sau về hiệu năng xử lý giao dịch và trả lời Câu hỏi.

Trong hình dưới đây, có một máy phục vụ gồm một bộ vi xử lý và hai c cứng. Cơ sở dữ liệu DB1 và DB2 nằm trên hai đĩa khác nhau. Trong trường hợp máy phục vụ này, các giao dịch khác nhau có thể thực hiện độc lập trên các đĩa. Bảng 1 đưa ra thời gian trung bình sử dụng tài nguyên hệ thống trong mỗi giao dịch dựa trên kết quả phân tích xử lý giao dịch.



Hình. Thiết kế máy phục vụ

Bảng 1 Thời gian trung bình sử dụng tài nguyên hệ thống trong mỗi giao dịch

Tài nguyên hệ thống	Thời gian sử dụng trung bình trong một giao dịch
CPU	20 ms
Disk 1	80 ms
Disk 2	40 ms

Số lượng giao dịch máy phục vụ xử lý trên một giây tại thời điểm bất kỳ gọi là TPS (Transcactions Per Second) của máy phục vụ đó. Khi hoạt động, TPS của máy phục vụ , tỉ lệ sử dụng của một tài nguyên, thời gian sử dụng trung bình của tài nguyên hệ thống trong mỗi giao dịch (viết tắt là thời gian sử dụn trung bình) liên hệ với nhau như sau::

Tỉ lệ sử dụng = TPS × Thời gian sử dụng trung bình ..... [1]

Thí dụ, TPS của máy phục vụ được tính theo công thức sau khi thời gian sử dụng trung bình của một tài nguyên hệ thống là 50ms và tỉ lệ sử dụng của tài nguyên đó là 80%.

$$\text{TPS} = \frac{0.8}{0.05} = 16$$

Câu hỏi

Từ các nhóm đáp án dưới đây, chọn đáp án đúng nhất để điền vào chỗ trống [ ] trong các mô tả sau.

(1) Khi giới hạn trên của tỉ lệ sử dụng mỗi tài nguyên hệ thống trong hình đạt 40%, giới hạn trên của TPS hệ thống là [A].

(2) Khi tiếp tục giả định giới hạn trên của tỉ lệ sử dụng mỗi tài nguyên hệ thống là 40%, để tăng giới hạn trên TPS hệ thống lên 4 lần thì cần giảm thời gian truy cập của đĩa đi ít nhất [B]. Giả sử tài nguyên hệ thống tăng lên n lần thì thời gian truy cập tài nguyên hệ thống trong xử lý giao dịch cũng giảm đi còn 1/n.

(3) Khi TPS của máy phục vụ là 10 thì thời gian đáp ứng trung bình trong xử lý một giao dịch xấp xỉ [C] lần sovới thời gian xử lý giao dịch đó. Ở đây, giả sử rằng thời gian đáp đáp ứng trung bình cho mỗi giao dịch bằng tổng thời gian đáp ứng trung bình cho mỗi tài nguyên hệ thống, và thời gian đáp ứng trung bình cho mỗi tài nguyên hệ thống được tính theo công thức sau đây.

$$\text{Thời gian đáp ứng trung bình} = \frac{\text{Thời gian sử dụng trung bình}}{1 - \text{Tỉ lệ sử dụng}} \quad \text{..... [2]}$$

(4) Theo công thức [1] và [2], có thể thấy khi TPS của máy phục vụ tăng lên, tỉ lệ sử dụng tài nguyên hệ thống cũng tăng và [D].

Nhóm đáp án cho A:

a) 0.005	b) 0.01	c) 0.02
d) 5	e) 10	f) 20

Nhóm đáp án cho B:

a) tăng số lượng Disk 1 lên bốn chiếc.
b) tăng số lượng Disk 2 lên bốn chiếc.
c) tăng số lượng Disk 1 lên hai và Disk 2 lên hai.
d) tăng số lượng Disk 1 lên hai và Disk 2 lên bốn.
e) tăng số lượng Disk 1 lên bốn và Disk 2 lên hai.

Nhóm đáp án cho C:

a) 0.48 b) 0.95 c) 1.66
d) 2.86 e) 3.51

Nhóm đáp án cho D:

a) thời gian đáp ứng trung bình tăng.
b) thời gian đáp ứng trung bình giảm.
c) thời gian sử dụng trung bình tăng.
d) thời gian sử dụng trung bình giảm.

Đáp án câu 1

Hiệu năng xử lý giao dịch

Đáp án đúng

A – d, B – e, C – e, D – a

Giải thích

Chủ đề của câu hỏi này là về hiệu năng tính toán trong xử lý giao dịch, nhưng nói chung, những tính toán về tỉ lệ sử dụng và thời gian đáp ứng trung bình đều nằm trong lý thuyết hàng đợi. Tuy vậy, các công thức tính, các thủ tục đã được mô tả cụ thể trong câu hỏi do vậy cũng không cần kiến thức về lý thuyết hàng đợi để trả lời câu hỏi này. Mục đích câu hỏi này là để xác định bạn có thể thực hiện tính toán chính xác dựa trên những giải thích của câu hỏi không.

(1) Liên quan đến TPS, số lượng giao dịch trong một giây, câu hỏi đưa ra thí dụ một TPS = 0.8 / 0.05 = 16 với giả thiết về thời gian sử dụng trung bình là 50 ms và tỉ lệ sử dụng là 80%. Bây giờ, giả sử rằng tỉ lệ sử dụng là 40%, và thời gian sử dụng trung bình cho mỗi thiết bị được cho trong bảng 1, ta có thể tính TPS cho mỗi thiết bị sử dụng giá trị trên và có kết quả sau.

Nguồn tài nguyên hệ	TPS
CPU	0.4 / 0.02=20
Disk 1	0.4 / 0.08=5
Disk 2	0.4 / 0.04=10

Nếu một quá trình có thể thực hiện song song với nhiều đơn vị xử lý, hiệu năng xử lý của toàn hệ thống có thể bị ảnh hưởng xấu đi bởi đơn vị xử lý có hiệu năng kém nhất. Những đơn vị xử lý như vậy gọi là cổ chai. Tr trường hợp này, Đĩa 1 là cổ chai, và giới hạn trên cho TPS của máy phục vụ là đáp án (d) 5..

(2) Để tăng gấp bốn lần giới hạn trên cho TPS của máy phục vụ, nghĩa là tăng lên 20. TPS của CPU là 20, của các đĩa 1 và 2 là 5 và 10, đều dưới ngưỡng 20. Do vậy cần cải thiện hiệu năng của các đĩa. Chiến lược cải thiện hiệu năng là tăng số lượng tài nguyên hệ thống (đĩa cứng). Hiệu năng của mỗi đơn vị đĩa tỉ lệ thuận với số lượng đĩa, bởi giả định rằng nếu số lượng tài nguyên tăng lên n lần thì thời gian truy cập giả đi còn 1/n. Một vài ý trong

câu hỏi có thể rắc rối, nhưng chúng ta có thể giả định rằng thời gian truy cập bằng với thời gian sử dụng trung bình (Thời gian truy cập = tỉ lệ sử dụng / TPS).

Trước hết, để tăng TPS của Disk 1 lên 20 trong khi giữ giới hạn trên 40% tỉ lệ sử dụng của nó, thời gian truy cập cần là 0.02 giây (= 0.4 / 20). Thời gian truy cập này bằng ¼ thời gian truy cập hiện tại – 0.08 giây, do vậy số lượng đĩa phải tăng lên gấp 4. Tương tự với Disk 2, thời gian truy cập hiện tại là 0.04 giây và cần giảm xuống 0.02, do vậy cần tăng gấp đôi số lượng Disk 2. Vậy đáp án đúng là (e), tăng số lượng Disk 1 lên bốn và số Disk 2 lên hai.

(3) Thời gian xử lý giao dịch là toàn bộ thời gian (tổng thời gian) cần thiết cho mỗi quá trình con trong giao dịch đó, không bao gồm thời gian đợi. Nó cách khác, thời gian đáp ứng trung bình bao gồm tổng thời gian xử lý giao dịch và thời gian đợi. Nội dung của quá trình xử lý giao dịch trong câu hỏi này không được chỉ ra, nhưng câu hỏi chỉ ra rằng "thời gian đáp ứng trung bình xử lý giao dịch là tổng thời gian đáp ứng trung bình cho mỗi tài nguyên hệ thống". Do vậy , có thể giả định thời gian xử lý giao dịch là tổng thời gian sử dụng trung bình của mỗi tài nguyên hệ thống. Như vậy, theo bảng 1, thời gian giao dịch là 20 + 80 + 40 = 140 ms. Thời gian đáp ứng trung bình cần được tính cho mỗi tài nguyên hệ thống, sử dụng công thức [1] và [2].

CPU: Tỉ lệ sử dụng = 10 × 0.02=0.2,
Thời gian đáp ứng trung bình = 0.02 / (1 – 0.2) = 0.025 = 25 ms.
Đĩa 1: Tỉ lệ sử dụng = 10 × 0.08=0.8,
Thời gian đáp ứng trung bình = 0.08 / (1 – 0.8) = 0.4 = 400 ms.
Đĩa 2: Tỉ lệ sử dụng = 10 × 0.04=0.4,
Thời gian đáp ứng trung bình = 0.04 / (1 – 0.4) = 0.067 = 67 ms.

Thời gian đáp ứng trung bình cho cả máy phục vụ là 25 + 400 + 67 = 492 ms. Câu hỏi là thời gian này lớn gấp bao nhiêu lần thời gian xử lý giao dịch, đáp án là 492 / 140 = 3.514, tương ứng với phương án (e).

(4) Từ công thức [1], Tỉ lệ sử dụng = TPS × Thời gian sử dụng trung bình, có thể thấy khi TPS tăng, tỉ lệ sử dụng cũng tăng. Công thức [2] như sau:

$$\text{Thời gian đáp ứng trung bình} = \frac{\text{Thời gian sử dụng trung bình}}{}$$

$$\frac{\text{Thời gian sử dụng trung bình}}{1 - \text{Tỉ lệ sử dụng}}$$

Từ công thức này chúng ta thấy, khi tỉ lệ sử dụng tăng lên, thời gian đáp ứng trung bình cũng tăng. Do đó, câu trả lời đúng là (a). Nếu các công thức trên chưa đủ thuyết phục bạn, hãy xem xét tình huống cụ thể sau : tỉ lệ sử dụng tăng lên từ 0.2 lên 0.4. Thời gian sử dụng trung bình có thể bất kỳ, ở đây ta chọn là 1. Nếu tỉ lệ sử dụng là 0.2, thời gian đáp ứng trung bình là 1 / (1 - 0.2) = 1.25. Nếu tỉ lệ sử dụng tăng lên 4, thời gian đáp ứng trung bình là 1 / (1 - 0.4) = 1.67.

[Hàng đợi]

Trả lời câu hỏi này không nhất thiết phải có những kiến thức về hàng đợi. Thực tế, câu hỏi trên đã gợi ý sử dụng lý thuyết hàng đợi để trả lời. Tuy đây là một bài tập, nhưng nó đưa ra ý tưởng khá hay để học lý thuyết hàng đợi. Phần sau đây sẽ giải thích thêm để bạn đọc tham khảo.

Chắc hẳn ai cũng đã từng chờ đợi ở chợ, siêu thị hay ngân hàng. Một khái niệm tương tự cũng được áp dụng cho xử lý máy tính. Người ta đánh giá hiệu năng của máy tính bằng hai phương diện: thời gian xử lý theo phương diện máy tính và thời gian đáp ứng theo phương diện người dùng. Lý thuyết hàng đợi áp dụng cho phương diện sau: thời gian đáp ứng.

Nhắc lại rằng, thời gian đáp ứng là thời gian từ khi yêu cầu được gửi đến khi hệ thống phản hồi yêu cầu đó. Trong thế giới thực, tại quầy thu ng trong ngân hàng, thời gian đáp ứng được tính từ lúc bạn xếp hàng đợi đến lúc kết thúc công việc. Thời gian này bao gồm thời gian hệ thống phục vụ những người đứng trước bạn và thời gian hệ thống phục vụ bạn.

Thời gian đáp ứng = thời gian đợi
+ thời gian bạn được phục vụ.

Thời gian bạn phải đợi những người khác gọi là thời gian đợi, và thời gian đáp ứng là tổng thời gian bạn phải đợi người khác và thời gian hệ thống phục vụ chính bạn. Thời gian đợi phụ thuộc vào chiều dài của hàng đợi và thời gian cần thiết phục vụ mỗi người. Thời gian phục vụ mỗi người có thể xác định bằng các thông số như kỹ năng...Thực tế, một câu hỏi về lý thuyết hàng đợi thường ước lượng trước được chiều dài hàng đợi.

Xem xét các hàng đợi thanh toán ở siêu thị. Khi nào chúng dài ra ? Chỉ khi có đông người đến mua sắm. Nhưng ngay cả khi có hai hàng đợi với cùng số lượng người và được thành lập cùng một lúc, thì người thu tiền chưa có kinh nghiệm sẽ mất nhiều thời gian hơn, hàng đợi sẽ dài hơn. Nói cách khác, thời gian đợi của hàng đợi được xác định bằng số lượng khách hàng và kỹ năng của người thu tiền. Thuật ngữ có thể diễn tả được cả hai thông tin trên là tỉ lệ sử dụng. Tỉ lệ sử dụng biểu diễn mức độ, khả năng sử dụng được của hệ thống, do vậy giá trị của nó được tính từ kỹ năng của người thu tiền và số lượng khách hàng.

Dựa trên ý tưởng trên, nếu thời gian sử dụng của CPU là 20 ms thì nó có thể xử lý 50 giao dịch / giây. Nếu có 10 giao dịch, tỉ lệ sử dụng là 10/50 = 0.2. Trong câu hỏi này, số lượng giao dịch trên một giây (TPS) được nhân với

thời gian sử dụng để tính tỉ lệ sử dụng. Giá trị TPS × (thời gian sử dụng) cho ta biết trong một giây tài nguyên được sử dụng bao lâu phần trăm. Đó cũng chính là tỉ lệ sử dụng. Tuy nhiên, hãy cẩn thận với đơn vị, đơn vị thời gian cho giao dịch là 1 giây, vì vậy thời gian sử dụng cũng phải tính theo giây. Nếu tất cả được chuyển đổi sang mili giây, chúng ta có TPS × (thời gian sử dụng) = 10 × 20 = 200 không hợp lý lắm với ý nghĩa của tỉ lệ sử dụng (không lớn hơn 1).

Trong các mô hình chung được sử dụng, chiều dài của hàng đợi được tính bằng (tỉ lệ sử dụng) / (1- tỉ lệ sử dụng). Vì vậy số lượng xử lý trong hàng đợi phương trình này có thể sử dụng để tìm chiều dài hàng đợi. Thời gian xử lý cả hàng đợi này là thời gian đợi. Do vậy có thể tính thời gian đợi = (chiều dài hàng đợi) × thời gian xử lý(thời gian sử dụng). Trong thí dụ trên, thời gian sử dụng là 20 mili giây và tỉ lệ sử dụng là 0.2. Thay số vào phương trình ta có chiều dài hàng đợi là 0.2 / (1 – 0.2) = ¼. Bạn có thể băn khoăn giá trị này không phải là số nguyên, nhưng có thể hiểu là cứ bốn người thì sẽ có một người phải đợi. Vậy thời gian đợi là ¼ thời gian xử lý, ¼ × 20 mili giây = 5 mili giây. Cộng với thời gian xử lý cho một giao dịch là 20 mili giây ta có thời gian đáp ứng là 25 mili giây. Hơn nữa, công thức [2] cũng được giới thiệ trong câu hỏi.

$$\text{Thời gian đáp ứng trung bình} = \frac{\text{Thời gian sử dụng trung bình}}{1 - \text{Tỉ lệ sử dụng}}$$

Nếu thay thời gian sử dụng trung bình (20 mili giây) và tỉ lệ sử dụng là 0.2 vào công thức, ta cũng có 20 / (1- 0.2) = 25 mili giây. Thời gian đáp ứng là tổng thời gian xử lý hàng đợi và thời gian xử lý một giao dịch. Ta có thể tính chiều dài hàng đợi như sau:

$$\text{Chiều dài hàng đợi} = \frac{\text{Tỉ lệ sử dụng}}{1 - \text{Tỉ lệ sử dụng}}$$

Ta có số lượng xử lý trong khoảng thời gian đáp ứng như sau::

$$\text{Số lượng xử lý trong khoảng thời gian đáp ứng} = \frac{\text{Tỉ lệ sử dụng}}{(1 - \text{Tỉ lệ sử dụng})} + 1$$

$$= \frac{\text{Tỉ lệ sử dụng}}{} + \frac{}{(1 - \text{Tỉ lệ sử dụng})}$$

$$= \frac{\frac{1 - \text{Tỉ lệ sử dụng}}{\text{Tỉ lệ sử dụng} + (1 - \text{Tỉ lệ sử dụng})}}{1 - \text{Tỉ lệ sử dụng}} = \frac{\frac{1 - \text{Tỉ lệ sử dụng}}{1}}{1 - \text{Tỉ lệ sử dụng}}$$

Và để tính thời gian đáp ứng, ta nhân số này với thời gian xử lý (thời gian sử dụng cho một giao dịch)

$$\frac{\text{Thời gian sử dụng}}{1 - \text{Tỉ lệ sử dụng}}$$

Tại sao lại có thời gian đợi ngay cả khi tỉ lệ sử dụng nhỏ hơn 1. Thí dụ, với Đĩa 1 trong câu hỏi này, tỉ lệ sử dụng là 0.8 khi TPS là 10. Nếu chúng ta tính chiều dài của hàng đợi, ta sẽ có 0.8 / (1 – 0.8) = 4. Thời gian truy cập đĩa là 80 mili giây với TPS là 10, thí dụ cứ 100 mili giây mới có một giao dịch, do vậy đĩa sẽ có 20 mili giây rảnh rỗi sau mỗi lần truy cập và bạn thường bỏ qua thời gian đợi này. Đây là dòng suy nghĩ đúng. Ở đây, giả định rằng thời gian truy cập và giao dịch xảy ra đều đặn, do vậy sẽ không có thời gian đợi. Tuy vậy mô hình của hàng đợi dựa trên giả thuyết thời gian truy cập cũng như các giao dịch xảy ra ngẫu nhiên. Nếu chúng là ngẫu nhiên hoàn toàn thì sẽ không thể thực hiện một tính toán nào, vì vậy giả định rằng ít nhất chúng ta biết được giá trị "trung bình". Để ý rằng tất cả các giá trị trong câu hỏi đều mang nghĩa "trung bình". Đây là một từ khóa, thời gian sử dụng là thời gian sử dụng trung bình, thời gian đáp ứng là thời gian đáp ứng trung bình. Trong phần giải thích trên, từ "trung bình" bị bỏ qua, nhưng nói một cách chính xác: tất cả phải có chữ "trung bình". Trong câu hỏi này, tất cả các mô hình hàng đợi, số lượng giao dịch trong một đơn vị thời gian được giả định có p Poisson, và thời gian xử lý một giao dịch có phân bố hàm mũ.

Câu hỏi 2

Q2. Đọc mô tả sau đây về lập lịch và trả lời câu hỏi.

Một nhà thầu được chọn để hoàn thành một dự án trong thời gian nhanh nhất và với chi phí thấp nhất.Dự án này có thể được chia làm tám công việc: từ A đến H, và mỗi công việc được gán một thứ tự thực hiện, như bảng 1 dưới đây. Một công việc sẽ không thể bắt đầu nếu nó yêu cầu một công việc khác trước đó mà chưa hoàn tất.

Bảng 2 đưa ra số lượng ngày công và chi phi nếu giao cho nhà thầu X hay Y. Nhà thầu X có thể hoàn tất công việc với chi phí ít hơn nhà thầu Y nhưng lại cần nhiều ngày công hơn.

Bảng 1 : Danh sách công việc

Công việc	Công việc yêu cầu hoàn thành
A	-
B	-
C	*A*
D	*B*
E	*B*
F	*C, D*
G	*E*
H	*F, G*

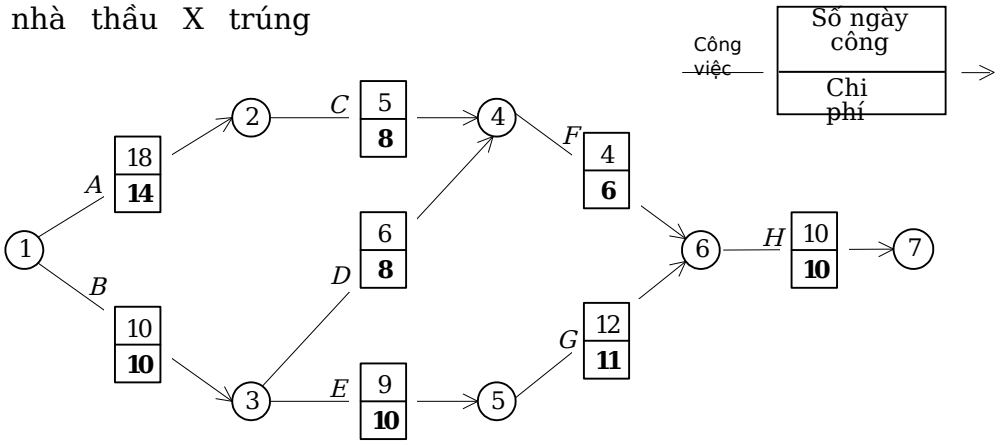
Bảng 2: Số lượng ngày công và chi phí theo các nhà thầu

Công việc	Nhà thầu *X*		Nhà thầu *Y*	
	Số lượng ngày công	Chi phí	Số lượng ngày công	Chi phí
A	18	14	16	15
B	10	10	8	13
C	5	8	4	10
D	6	8	5	10
E	9	10	8	13
F	4	6	3	9
G	12	11	10	12
H	10	10	9	12

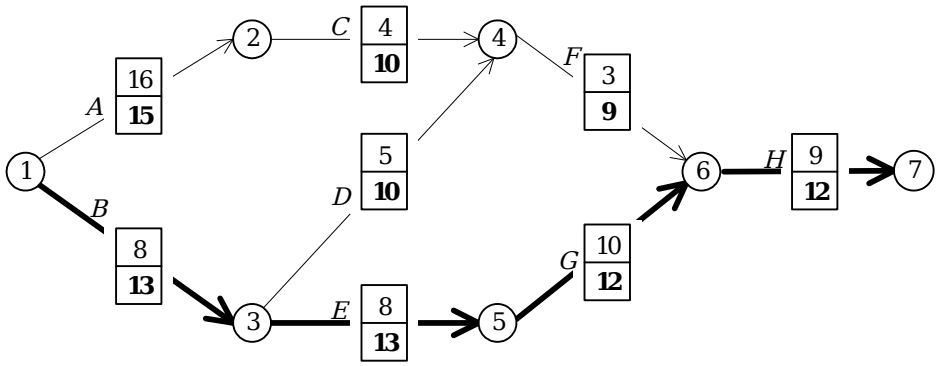
Trang tiếp theo mô tả cách thức các nhà thầu hoàn tất công toàn bộ công việc.

Mỗi sơ đồ nêu tên ở trang sau mô tả tương ứng cho tùy trường hợp nhà thầu X hoặc Y trúng thầu để hoàn tất công việc.

Nếu nhà thầu X trúng thầu



Nếu nhà thầu Y trúng thầu



Hình: sơ đồ nêu tên của dự án

Câu hỏi con

Từ các nhóm câu trả lời dưới đây, chọn câu trả lời đúng nhất điền vào chỗ trống [ ] chu phù hợp với các mô tả sau.

(1) Khi nhà thầu X trúng thầu, chi phí hoàn tất dự án là [A] và số lượng ngày công cần thiết là [B].

(2) Khi nhà thầu Y trúng thầu , chi phí hoàn tất dự án là 94 và số ngày công cần thiết là 35. Đường căng là đường tô đậm BEGH trong trên Nếu nhà thầy X trúng thấu để thực hiện một phần của dự án, số lượng ngày công vẫn như trên: 35 ngày, nhưng chi phí có t xuống, cụ thể, chi phí thấp nhất sẽ là [C].

Nhóm câu trả lời

a) 35 b) 41 c) 63 d) 74 e) 77

f) 86 g) 87 h) 88 i) 89 j) 90

Đáp án câu 2

Lập lịch

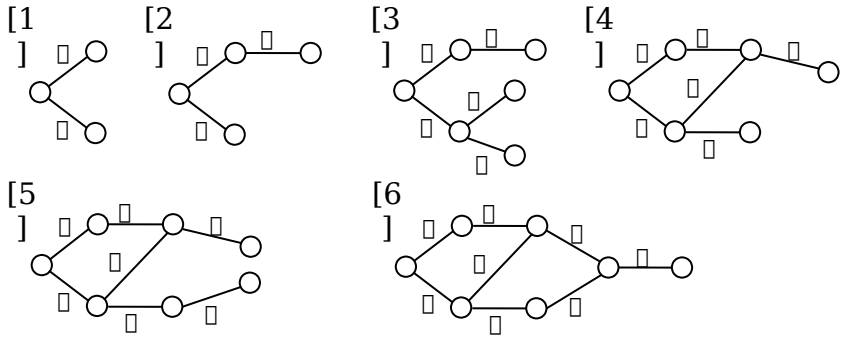
Đáp án đúng

A – e, B – b, C – g

Giải thích

Đây là câu hỏi liên quan đến việc lựa chọn nhà thầu sao cho dự án có thể hoàn thành với số ngày công tối thiểu và chi phí thấp nhất có thể.
Dự án này có thể chia làm tám công việc đánh dấu từ A đến H và chúng có yêu cầu về thứ tự thực hiện, như đã được chỉ ra trong bảng 1. Các hình dưới đây mô tả mỗi liên hệ này, từ đó xây dựng ra biểu đồ PERT.

[1] *A* và *B* không có yêu cầu công việc trước đó, do vậy có thể thực hiện song song.
[2] *C* cần *A* phải được thực hiện trước, vì vậy *C* phải thực hiện sau *A*.
[3] *D* và *E* cần B thực hiện trước, vì vậy chúng cần phải thực hiện sau *B*.
[4] *F* cần *C* và *D* thực hiện trước, vì vậy *F* phải được thực hiện sau *C* và *D*.
[5] *G* cần *E* thực hiện trước, vì vậy *G* phải thực hiện sau *E*.
[6] *H* cần *D* và *G* thực hiện trước, vì vậy *H* phải thực hiện sau *F* và *G*.



A: Chi phí thuê nhà thầu X thực hiện tất cả các công việc đơn giản là tổng tất cả chi phí trong bảng 2. Đó là (e): 14 + 10 + 8 + 8 + 10 + 6 + 11 + 10 = 77.

B: Số lượng ngày công cần thiết để hoàn tất có thể tính theo biểu đồ PERT. Số

lượng công việc khá ít do vậy sẽ không mất nhiều thời gian để xem xét tất cả các đường đi có thể trong biểu đồ.

- $A \rightarrow C \rightarrow E \rightarrow H$ (37 ngày)
- $B \rightarrow D \rightarrow F \rightarrow H$ (30 ngày)
- $B \rightarrow E \rightarrow G \rightarrow H$ (41 ngày)

Do vậy số lượng ngày công cần thiết để hoàn thành dự án là 31 + 10 = 41 ngày (b). Đường căng ưu là
$B \rightarrow E \rightarrow G \rightarrow H$.

C: Chi phí cần thiết để thuê nhà thầu Y thực hiện tất cả công việc là tổng các chi phí của Y trong bảng 2: 15 + 13 + 10 + 10 + 13 + 9 + 12 + 12 = 94, và số ngày cần thiết là 35. Mục tiêu bây giờ là để giữ nguyên 35 ngày nà nhưng giảm chi phi bằng việc thuê X thực hiện một phần dự án.

Từ câu hỏi và từ bảng 2, dễ thấy rằng mỗi công việc, nhà thầu X mất nhiều thời gian hơn nhưng chi phí thì ít hơn nhà thầu Y. Từ điều kiện này, có thể giảm chi phí nếu ta thuê X thực hiện những công việc có dư ra một chút thời gian trong dự án nhưng không ảnh hưởng đến các công việc sau đó.

Đường đi tối ưu là B – E – G - H, tất cả mất 35 ngày, ta cần gán những công việc không nằm trên đường đi này : A, C, D, và F cho nhà thầu X. Ở đây, tất cả những công việc cho tới nút [6] phải được hoàn tất trong 35 – (số ngày cần thiết cho H) = 35 – 9 = 26 (ngày). Hơn nữa, bởi số lượng ngày dư ra và chi phí tùy thuộc và các công việc, ta ưu tiên vào những công việc có tỉ lệ giữa chi phí/ngày công cao. Bảng dưới đây so sánh các công việc A,C,D và F giữa nhà thầu X và Y:

Công việc	# số ngày	Chi phí	Chi phí /Ngày
A	+2	-1	-0.5
C	+1	-2	-2
D	+1	-2	-2
F	+1	-3	-3

Trước hết giả sử cho nhà thầu X làm công việc F. Số lượng ngày tăng lên 1 nhưng chi phi giảm đi 3.

Nếu F được gán cho X, các công việc cho đến [4] phải kết thúc trong vòng 26 –(số ngày để X hoàn thành F) = 26 – 4 = 22 (ngày). Ở đây, nếu ta cũng thuê nhà thầu X thực hiện công việc C và D, mỗi công việc sẽ bị tăng lên 1 ngày, nhưng chúng ta vẫn có thể hoàn tất trong vòng 22 ngày, cùng lúc đó giảm chi phí đi 4 (mỗi công việc giảm đi 2).

Cuối cùng xem xét công việc A. Công việc C đã được gán cho nhà thầu X, tất cả các công việc đến nut [2] phải hoàn tất trong vòng 22- (số ngày X hoàn thành C) = 22 – 5 = 17 ngày. Tuy vậy, nếu công việc A được gán cho nhà thầu X, số ngày cần thiết đến điểm đó sẽ là 18, vì vậy không thể gán A cho X.

Như vậy, chi phí tối thiểu, khi nhà thầu X được thuê thực hiện các công việc C , D và F là

(chi phí thuê Y làm tất cả các công việc) – (chi phí giảm được khi thuê X)
= 94 – (3 + 2 + 2)
= 94 – 7 = 87.
Đáp án đúng là (g).

Câu hỏi 3

Q3. Đọc mô tả sau đây về biểu diễn số dấu phảy động, sau đó trả lời các Câu hỏi con từ 1 đến 3.

Biểu diễn một số dấu phảy động 32-bit như hình 1 dưới đây.

Ý nghĩa của các bit như sau:

(1) Bit 0: Dấu

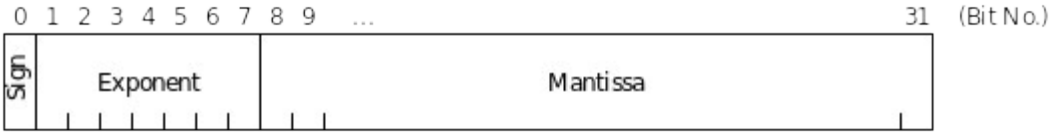
Bit này chưa dấu của phần định trịnh "0" biểu diễn giá trị dương, và "1" biểu diễn giá trị âm.

(2) Các bit 1 đến 7: Phần mũ.

Các bit này biểu diễn phần mũ theo hệ nhị phân. Giá trị âm được biểu diễn theo số bù hai.

(3) Các bit 8 đến 31: Phần định trị.

Những bit này biểu diễn giá trị tuyệt đối của phần định trị. Bit 8 là bit đầu tiên trong phần lẻ của số dấu phảy động.

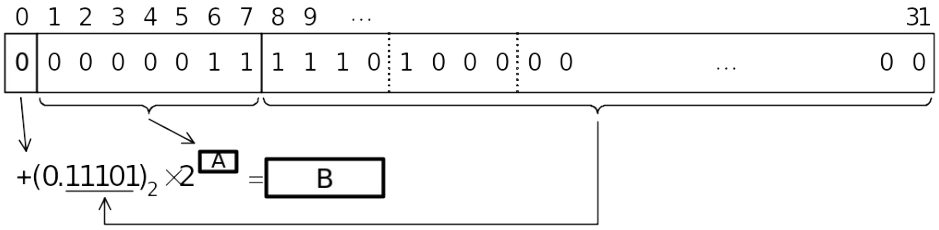


Hình .1 Định dạng biểu diễn số dấu phảy động

Câu hỏi con 1

Hình 2 minh họa phương pháp chuyển đổi giá trị nhị phân trong biểu diễn số dấu phảy động sang giá trị thập phân. Chọn giá trị thập phân điền vào các chỗ trống A và B từ các nhóm câu trả lời dưới đây.

"$(0.11101)_2$" trong hình 2 là một số nhị phân.



Hình. 2 Phương pháp chuyển đổi số nhị phân trong biểu diễn dấu phẩy động sang số thập phân.

Nhóm câu trả lời cho A:

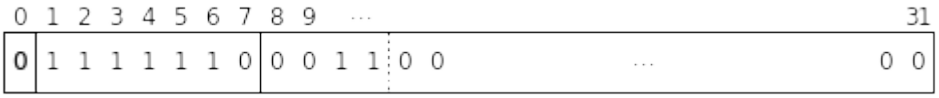
a) 0 b) 1 c) 2 d) 3 e) 4

Nhóm câu trả lời cho B:

a) 0.725 b) 0.74 c) 7.25
d) 7.4 e) 72.5 f) 74

Câu hỏi con 2

Từ nhóm câu trả lời dưới đây, hãy chọn ra biểu diễn thập phân đúng cho số nhị phân trong dạng dấu phảy động sau.



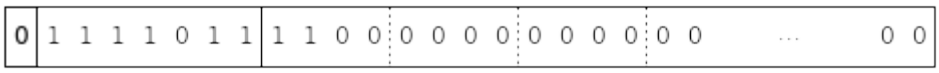
Nhóm câu trả lời:

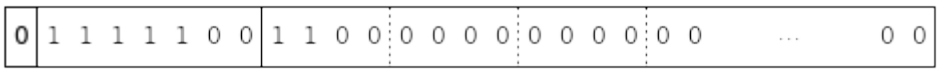
a) 3×2^{-6} b) 3×2^{-4} c) 3×2^{-1} d) 3×2^{0}
e) 3×2^{1} f) 3×2^{2} g) 3×2^{122}

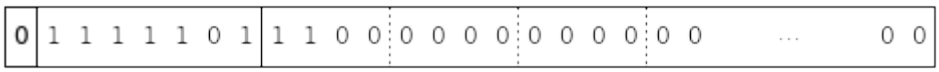
Câu hỏi con 3

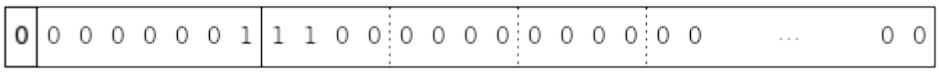
Chọn dạng biểu diễn “đã được chuẩn hóa” của số dấu phảy động trong Câu hỏi 2 từ nhóm câu trả lời dưới đây. Ở đây, "chuẩn hóa" nghĩa là biến đổi phần mũ và phần định trị để bit trái nhất của phần định trị là “1”.

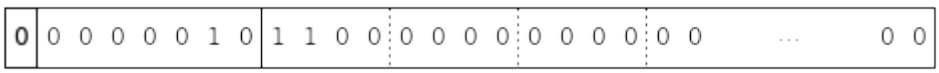
Nhóm câu trả lời:

a) 

b) 

c) 

d) 

e) 

Đáp án câu 3

Biểu diễn số dấu phảy động

Đáp án đúng

[Câu hỏi con 1] A – d, B – c

[Câu hỏi con 2] a

[Câu hỏi con 3] b

Giải thích

Đây là câu hỏi về biểu diễn dấu phảy động. Phần lớn các phần thi buổi sáng có câu hỏi về biểu diễn dấu phảy động. Tuy vậy, hầu hết người thi có vẻ yếu về phần này. Phần lớn câu hỏi về số dấu phảy động được đưa ra trong các kỳ thi FE có thể trả lời nếu hiểu rõ về việc chuyển đổi hệ cơ số và số dấu phảy động, do vậy bạn phải nắm được những khái niệm này.

Dấu phảy động là một cách biểu diễn giá trị số học bằng bậc của chúng (một số xấp xỉ kích thước của số đó, kiểu như “hàng nghìn”, “hàng triệu”) và các chữ số đầu của phần định trị. Trong biểu diễn dấu phảy tĩnh, số chữ số cần biểu diễn (số bit) phụ thuộc vào bậc của giá trị số học, nhưng bên trong máy tính xử lý tất cả các số đều theo chiều dài cố định, do vậy khoảng giá trị số học được biểu diễn bị giới hạn. Mặt khác, khi cần xử lý một khoảng rộng số giá trị, không phải tất cả các chữ số đều quan trọng như nhau, thường các chữ số đầu của phần định trị quan trọng hơn. Do đó người ta chấp nhận sai số để biểu diễn được khoảng các số giá trị học rộng hơn. Số dấu phảy động là một phương pháp biểu diễn số học đạt được yêu cầu trên, có khả năng biểu diễn một khoảng rộng các giá trị số (với những giá trị tuyệt đối rất lớn và rất nhỏ) bằng một số cố định chữ số (số bit).

Thí dụ : số thập phân (+)123

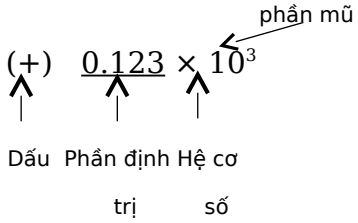
Khi di chuyển dấu phảy phần thập phân, ta có :

$$
\begin{aligned}
&: \\
&= 1230 \times 10^{-1} \\
(123)_{10} &= 123 \times 10^{0} \\
&= 12.3 \times 10^{1} \\
&: \\
&= \boxed{0.123 \times 10^{3}} \quad \ldots\ldots [1] \\
&= 0.0123 \times 10^{4}
\end{aligned}
$$

:

Như trên, bằng việc thay đổi vị trí dấu phảy thập phân, một giá trị số học có thể biểu diễn bằng vô số cách. Để thống nhất một chuẩn, người ta thiết lập ra một luật sử dụng để biểu diễn có dạng như [1]. Quá trình này gọi là chuẩn hóa. Chuẩn hóa là việc chuyển đổi để có nhiều chữ số có nghĩa bằng việc chuyển chữ số bên trái khác 0 về chữ số đầu tiên sau dấu phảy. Căn cứ vào hệ cơ số nào sử dụng (trong trường hợp này là 10), ta có thể xác định giá trị số học bởi dấu, phần định trị và phần mũ của số đó.

Trong thí dụ trên:



Bên trong máy tính, tất cả phần dấu (0 với số dương, 1 với số âm), phần định trị, và phần mũ đều được biểu diễn trong hệ nhị phân. Phần mũ có hai định dạng biểu diễn:

(1) Định dạng dấu phảy cố định (số bù hai)
(2) Biểu diễn dịch (biểu diễn vượt)

Định dạng thứ nhất không được sử dụng thường xuyên, nhưng các câu hỏi liên quan đến định dạng này vẫn có thể xuất hiện trong bài thi. Định dạng thứ hai được sử dụng thực tế trong máy tính. Định dạng này đã được giải thích kỹ trong câu hỏi.

Việc chuyển đổi được thực hiện như sau:

(1) Số thập phân được chuyển sang hệ nhị phân bằng phép chuyển cơ số..
(2) Chuẩn hóa số nhị phân sao cho chữ số đầu tiên khác không là chữ số đầu tiên sau dấu phảy.
(3) Biểu diễn phần mũ theo số bù hai.
(4) Dấu được đặt vào phần đầu, mũ được đặt vào phần mũ và phần định trị được đặt vào phần định trị.

Với số thập phân trong thí dụ trên, để ý rằng

$(123)_{10} = (1111011)_2$

Và dạng biểu diễn dấu phảy động của nó là:

$(+)\ (1111011)_2 \times 2^0 = (+)(0.1111011)_2 \times 2^7$

[Câu hỏi con 1]

Bởi vì phần mũ được biểu diễn dưới dạng số bù 2, (0000011_2) = "+3". Do đó, giá trị biểu diễn bởi hình 2 là

$(+)\ (0.11101)_2 \times 2^3$
$= (111.01)_2 \times 2^0$

$= 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$

$= (7.25)_{10}$

Đáp án đúng cho chỗ trống A là (d), đáp án đúng cho chỗ trống B là (c).

[Câu hỏi con 2]

Từ biểu diễn bit $(1111110)_2$ của phần mũ, ta thấy bit đầu tiên là 1, do đó giá trị phần mũ là âm. Đây là số bù hai 7 bit (định dạng dấu phảy tĩnh), do đó giá trị là "-2". Giá trị được cho trong câu hỏi con này là "$(+) (0.0011)_2 \times 2^{-2}$", có thể viết lại như sau.

$(+) (0.0011)_2 \times 2^{-2}$
$= (+) ((11)_2 \times 2^{-4}) \times 2^{-2}$
$= (+) (11)_2 \times 2^{-6}$
$= (+) (3)_{10} \times 2^{-6}$

=> Đáp án đúng là (a).

[Câu hỏi con 3]

Như đã giải thích ở trên, chuẩn hóa là thao tác dịch chuyển chữ số k
không đầu tiên bên trái đến vị trí đầu tiên của phần thập phân. Do đó, số đã cho trong câu hỏi con 2 có thể biểu diễn lại như sau.

$(+) (0.0011)_2 \times 2^{-2}$
$= (+) ((0.11)_2 \times 2^{-2}) \times 2^{-2}$
$= (+) (0.11)_2 \times 2^{-4}$

Đáp án đúng sẽ là số nào có phần mũ biểu diễn trong dạng bù hai của “-4”. “+4” là $(00000100)_2$, số bù 2 của nó là $(11111100)_2$. Vậy đáp án đúng là (b). Phần mũ trong các lựa chọn khác lần lượt là (a) -5, (c) -3, (d) +1, và (e) +2.

Thuật toán

6

[Phạm vi câu hỏi]
sắp xếp, tìm kiếm, xử lý chuỗi ký tự, xử lý file, biểu đồ, đồ thị, tính toán số, ….

Câu hỏi 1

Q1. Đọc mô tả chương trình và chương trình sau đây, sau đó trả lời các câu hỏi con.

[Mô tả chương trình]

Hàm `RadixConv` là một chương trình chuyển đổi một xâu ký tự số trong hệ `M` (2 <= `M` <= 16) thành một xâu ký tự số trong hệ `N` (2 <= `N` <= 16).

(1) Một xâu ký tự hệ M bao gồm M ký tự số không có ký tự trắng. Trong trường hợp các hệ từ 11-16, các ký tự "A" ~ "F" được sử dụng để ký hiệu các giá trị từ 10 ~ 15.

(2) Hàm `RadixConv` đầu tiên chuyển một xâu ký tự số hệ M thành một số nguyên (integer) sau đó sẽ chuyển sang một xâu ký tự số hệ N. Hà `MToInt` chuyển một xâu ký tự số hệ M thành một số nguyên và hà `IntToN` chuyển số nguyên thành một xâu ký tự số hệ N.

(3) Hàm `MToInt` và hàm `IntToN` sử dụng các hàm sau:

(i) Hàm `ToInt` chuyển một ký tự số ("0", "1", ..., or "F") thành một số nguyên.

(ii) Hàm `ToStr` chuyển một số nguyên `Q` (0 <= `Q` <= 15) thành một ký tự số ("0", "1", ..., "F")

(iii) Hàm có sẵn `Length` trả về chiều dài của một chuỗi.

(iv) Hàm có sẵn `Substr` lấy ra một phần của chuỗi.

(4) Các bảng từ 1 đến 5 dưới đây đưa ra danh sách các đối số và giá trị trả về của các hàm.

Bảng 1 `RadixConv`

Đối số/ Giá trị trả về	Kiểu dữ liệu	Ý nghĩa
`Frdx`	Integer	Cơ số của xâu ký tự số trước khi chuyển đổi (2<= `Frdx` <= 16)
`Fnum`	Char	Xâu ký tự số trước khi chuyển
`Trdx`	Integer	Cơ số của chuỗi số sau khi chuyển (2 <= `Trdx` <= 16)
Giá trị trả về	Char	Xâu ký tự số sau khi chuyển trong hệ `Trdx`

Bảng 2 `MToInt`

Đối số/ Giá trị trả về	Kiểu dữ liệu	Ý nghĩa
`Rdx`	Integer	Cơ số của xâu ký tự số trước khi chuyển đổi (2 <= `Rdx` <= 16)
`Num`	Char	Xâu ký tự số trước khi chuyển
Giá trị trả về	Integer	Số nguyên sau khi chuyển

Bảng 3 IntToN

Đối số/ Giá trị trả về	Kiểu dữ liệu	Ý nghĩa
Val	Integer	Số nguyên trước khi chuyển
Rdx	Integer	Cơ số của chuỗi số sau khi chuyển đổi (2 <= Rdx <= 16)
Giá trị trả về	Char	Xâu ký tự số sau khi chuyển về hệ cơ số Rdx

Bảng 4 ToInt

Đối số/ Giá trị trả về	Kiểu dữ liệu	Ý nghĩa
P	Char	Các ký tự số trước khi chuyển đổi ("0", "1", ..., or "F")
Giá trị trả về	Integer	Số nguyên sau khi chuyển

Bảng 5 ToStr

Đối số/ Giá trị trả về	Kiểu dữ liệu	Ý nghĩa
Q	Integer	Số nguyên trước khi chuyển đổi (0 <= Q <= 15)
Giá trị trả về	Char	Ký tự số sau khi chuyển đổi.

[Chương trình]

```
○ char_type: RadixConv (int_type: Frdx, char_type: Fnum, int_type: Trdx)
• return IntToN(MToInt(Frdx, Fnum), Trdx)
                          /* Lấy giá trị của IntToN là giá trị trả về của hàm */

○ int_type: MToInt(int_type: Rdx, char_type: Num)
○ int_type: Idx, Val
• Val ← 0
■  Idx: 1, Idx <= Length(Num), 1  /* Hàm Length trả về độ dài của Num */
   □Val ← [   A   ] + ToInt(Substr(Num, Idx, 1))
■     /* Hàm Substr trả về Idx(>= 1) ký tự bắt đầu từ vị trí Num */
• return Val                  /* Lấy Val là giá trị trả về của hàm */

○ char_type: IntToN(int_type: Val, int_type: Rdx)
○ int_type: Quo               /* Quotient: Thương số */
○ int_type: Rem               /* Remainder: Số dư */
○ char_type: Tmp
• [   B   ]
• Tmp ← ""
■ Quo >= Rdx
  • Rem ← Quo % Rdx
  • Tmp ← ToStr(Rem) + Tmp /* + là toán tử nối chuỗi */
  • [   C   ]
■
• [   D   ]
• return Tmp                  /* Lấy Tmp là giá trị trả về của hàm */

○ int_type: ToInt(char_type: P)
○ int_type: Idx
○ char_type: Code[16]         /* Chỉ số bắt đầu từ 0 */
/* Code chứa các giá trị khởi tạo "0"□"1"□"2"□"3"□"4"□"5"□"6"□"7"□     */
/* "8"□"9"□"A"□"B"□"C"□"D"□"E"□"F"  theo thứ tự này                    */
/* Giá trị các ký tự được tăng dần                                     */
• Idx ← 0
■ [   E   ]                   /* So sánh chuỗi */
  • Idx ← Idx + 1
■
• return Idx                  /* Lấy Idx là giá trị trả về của hàm */

○ char_type: ToStr(int_type: Q)
○ char_type: Code[16]         /* Chỉ số bắt đầu từ 0 */
/* Code chứa các giá trị khởi tạo  "0"□"1"□"2"□"3"□"4"□"5"□"6"□"7"□
       */
/* "8"□"9"□"A"□"B"□"C"□"D"□"E"□"F"  theo thứ tự này                    */
```

```
/* Giá trị các ký tự tăng dần theo thứ tự này                    */
• return Code[Q]              /* Lấy Code[Q] là giá trị trả về của hàm */
```

Câu hỏi con

Từ các nhóm đáp án phía dưới, chọn đáp án đúng để điền vào các ô trống ☐ trong chương trình trên.

Nhóm đáp án cho A:

a) `Rdx`
b) `Val`
c) `Val * Rdx`
d) `Val / Rdx`

Nhóm đáp án cho B, C và D:

a) `Quo ← Quo / Rdx`
b) `Quo ← Quo / Rem`
c) `Quo ← Rdx`
d) `Quo ← Rem / Rdx`
e) `Quo ← Val`
f) `Rem ← Rdx`
g) `Rem ← Val`
h) `Tmp ← ToStr(Quo) + Tmp`
i) `Tmp ← ToStr(Rem) + Tmp`

Nhóm đáp án cho E:

a) `P < Code[Idx]`
b) `P > Code[Idx]`
c) `P <= Code[Idx]`
d) `P >= Code[Idx]`

Đáp án câu 1

Chương trình chuyển đổi cơ số (Giả ngôn ngữ)

Đáp án đúng

A – c, B – e, C – a, D – h, E – b

Chú thích

Trong [Chương trình], có 5 hàm con được đưa ra tuần tự. Dạng câu hỏi này khác so với các câu hỏi trước, nên cần được chú ý. Tuy nhiên, điều quan trọng là tiếp cận chương trình bằng cách chia nó thành các phần nhỏ. Cụ thể cách tiếp cận với dạng câu hỏi này như sau.

Mọi chương trình, không quan tâm nó lớn như thế nào, đều được tạo ra bởi sự kết hợp của các hàm cần thiết nhỏ hơn để hoàn thành toàn bộ chức năng. Một chương trình có chức năng càng phức tạp thì càng cần chia thành các hàm con, từ các hàm con tạo nên một chương trình lớn. Tuy nhiên, nếu bạn chỉ tập trung vào các hàm con thì bạn chỉ cần suy nghĩ về nó mà thôi không quan tâm tới kích thước toàn bộ chương trình. Thông thường, khi nghĩ về một câu hỏi phức tạp, gợi ý là bắt đầu bằng việc bỏ đi nguyên nhân gây ra sự phức tạp. Khi kích thước là nguyên nhân thì bạn nên chia thành các phần nhỏ hơn. Khi một hàm là phức tạp, bạn nên nghĩ tới các hàm nhỏ hơn cần thiết kết hợp với nhau để hoàn thành chức năng của hàm đó. Tuy nhiên chúng ta đã được học trong mục liên quan tới tính chất cố kết và kết nối giữa các module, thiết kế chương trình là cân nhắc lợi hại khi chia thành các module (Các hàm nhỏ hơn) có tính độc lập cao so với nhau. Trong các câu hỏi thi có các chương trình gần như ngược lại với nguyên tắc thiết kế, bạn không phải lo lắng nhiều về điều đó.

Đầu tiên, cần phải hiểu chức năng của toàn bộ chương trình. Tại điểm bắt đầu của [Mô tả chương trình], chức năng đã được mô tả rõ ràng như sau: "Hàm `RadixConv` là một chương trình chuyển xâu ký tự số hệ M thành xâu ký tự số hệ N". Chúng ta nên chắc chắn nhớ chức năng này trong đầu khi trả lời các câu hỏi dùng vào “xem rừng tìm cây”. Nhưng đồng thời cẩn thận đừng quá trung vào “rừng” toàn bộ chương trình.

Tiếp theo, chúng ta cố gắng hiểu tổng quát các hàm nhỏ. Mỗi câu hỏi luôn luôn có gợi ý, bạn phải tìm ra nó. Nếu một chương trình được phân chia rõ ràng thành các thủ tục hoặc các hàm nhỏ hơn, gợi ý thường ở trong các tham s hoặc giá trị trả về. Hãy nhớ chức năng của toàn bộ chương trình, sau đây chúng ta sẽ xem xét tổng quát mỗi hàm.

- **RadixConv** (Bảng 1)
  Chính hàm này là mục đích của toàn bộ chương trình lớn. Chúng ta hãy xem cách tổ chức và các thành phần liên quan. Các tham số là cơ số của chuỗi ký tự trước khi chuyển (Frdx), xâu ký tự số trước khi chuyển (Fnum) và cơ số của xâu ký tự số sau khi chuyển (Trdx). Giá trị trả về là xâu ký tự số đã được chuyển đổi. Chúng ta thấy rằng phần này là chuyển một xâu ký tự số (Fnum) trong hệ cơ số M (Giá trị của Frdx) thành xâu ký tự số trong hệ cơ số N và trả về xâu ký tự số là kết quả của việc chuyển đổi. Đây chính là mục đích chính của toàn bộ chương trình.

- **MToInt** (Bảng 2)

  Các tham số gồm có cơ số của xâu ký tự số trước khi chuyển (Rdx) và xâu ký tự số trước khi chuyển (Num); giá trị trả về là số nguyên đã chuyển. Do đó đây là hàm để chuyển một xâu ký tự số (Num) thành một số nguyên (Giá trị số). Do kiểu dữ liệu của các tham số là xác định, xâu ký tự số là thuộc kiểu kí tự. Thông thường, các xâu ký tự số không thể được sử dụng cho tính toán, vì vậy các phép tính toán cần chuyển các xâu ký tự số thành các giá trị số tương ứng. Mặc khác, tên của hàm "MToInt" cũng gợi ý "Chuyển từ M thành số nguyên", nên có thể đoán được chức năng của hàm từ tên của nó.

- **IntToN** (Bảng 3)

  Các tham số gồm có số nguyên trước khi chuyển (Val), cơ số của xâu ký tự số sau khi chuyển (Rdx), và giá trị trả về là xâu ký tự số sau khi chuyển. Dựa trên các thông tin này thì đây là hàm chuyển số nguyên trước chuyển (Val) thành một xâu ký tự số trong hệ cơ số N. Tên "IntToN" có thể hiểu nghĩa là "Chuyển từ số nguyên thành N”.

- **ToInt** (Bảng 4)

  Tham số là một ký tự số (P) trước khi chuyển, và giá trị trả về là số nguyên sau khi chuyển. Ký tự số (P) được chuyển thành một số nguyên. Chú ý rằng ký tự số được coi như là một ký tự (xâu ký tự).

- **ToStr** (Bảng 5)

  Tham số là số nguyên (Q) trước khi chuyển, và giá trị trả về là một ký tự số đã chuyển. Hàm chuyển một số nguyên (Q) thành một ký tự số. Bạn có thể lo lắng bởi việc không có cơ số trong chuyển đổi. Tuy nhiên, dù bạn sử dụng cơ số nào đi nữa, giá trị tương ứng với ký tự số là giống nhau. Cơ số được sử dụng để quyết định phạm vi biểu diễn của một giá trị số mà tham số Q nắm giữ. Ví dụ, trong hệ cơ số 2, có các giá trị 0 hoặc 1; trong hệ cơ số 16, có phạm vi từ 0 tới 15. Trong các trường hợp khác, các giá trị số giống nhau tương ứng với một số.

    Khi kiểm tra các hàm này, bạn có lẽ phân vân về cơ số sử dụng cho số nguyên trước và sau khi chuyển. Tuy nhiên, các giá trị số là giá trị không phụ thuộc vào cơ số. Khi chúng ta nói "hệ cơ số n", chúng ta nói tới hệ cơ số mà trong đó ta biểu diễn các giá trị số (Nếu bạn phải nghĩ về cơ số trong trường hợp này, chọn 2 bởi vì mọi thứ trong máy tính được biểu diễn trong hệ nhị phân). Khi chuyển một xâu ký tự số trong hệ 7 sang hệ 9, chúng ta chuyển tạm thời sang hệ 10; tuy nhiên, chúng ta không chuyển thực tế sang

hệ 10 mà là chuyển thành một giá trị số. Hệ đếm dễ dàng nhất cho chúng ta là hệ cơ số 10, vì thế chúng ta chuyển số sang hệ cơ số 10 để hiểu giá trị của nó.

Ô trống A:

Hàm "MToInt" là hàm chuyển một xâu ký tự số trong hệ cơ số M thành một số nguyên. Từ dòng cuối "return Val" chúng ta nhìn thấy giá trị số nguyên sau khi chuyển đổi được lưu trữ trong Val. Bản thân hàm là một vòng lặp của quá trình có điều kiện là "Idx <= Length(Num)". Giá trị khởi tạo của Idx là "Length(Num)" chỉ ra độ dài của xâu ký tự số Num, và được tăng dần lên 1 đơn vị, do đó quá trình lặp có số lần lặp bằng số "Substr(Num, Idx, 1)" ở trong vòng lặp dùng để "Lấy ra một ký tự tại vị trí thứ Idx(Tính từ đầu chuỗi) của xâu ký tự Num". Chương trình lấy một ký tự tại một thời điểm từ đầu của xâu ký tự số Num và cộng giá trị vào biến Val, biến chứa kết quả của quá trình chuyển đổi.

Việc chuyển đổi một số hệ M thành một số giá trị nguyên tương đương với việc chuyển sang hệ 10. Nhớ lại cách thực hiện chuyển đổi. Ví dụ, xem xét một số nhị phân 101. Tính toán dẫn đến $1\times2^2(=4) + 0\times2^1(=0) + 1\times2^0(=1)$, cho giá trị 5 (trong hệ 10). Nói cách khác, chúng ta lấy giá trị số của mỗi ký tự số nhân với trọng số tương ứng với vị trí của bit đó và tính ra kết quả. Nhưng nhóm trả lời lại có vẻ như không chứa cách này. Chú ý là $1\times2^2 + 0\times2^1 + 1\times2^0 = ((1\times2)+0)\times2+1$, vì thế cách sau cùng là cách được dùng ở đây. "2" tương ứng với M trong hệ cơ số M, vì thế chúng ta có thể nhân Val với Rdx (Biến chứa giá trị M) và cộng giá trị số tại mỗi vị trí các bit. Bởi vậy, đáp án đúng là (c). Nhớ lại là hàm MToInt dùng để chuyển một ký tự số thành một số nguyên (giá trị).

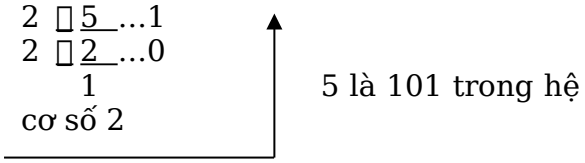
Nếu bạn chú ý là $1\times2^2 + 0\times2^1 + 1\times2^0 = ((1\times2) + 0)\times2+1$, câu trả lời sẽ không khó để tìm ra. Thậm chí nếu bạn không chú ý điều đó, bạn cũng không cần phải bỏ qua việc tìm ra câu trả lời. Sử dụng thực tế là đáp án đúng nằm trong nhóm các đáp án. Khi tham số Rdx = 2 và Num = "101" được dùng cho hàm này, có thể biết Val là 5. Khi gán mỗi lựa chọn trả lời vào trong ô trống và kiểm tra chương trình, đáp án nào cho kết quả đúng là đáp án đúng. Trong ví dụ, Num có 3 ký tự số nên quá trình thực hiện 3 lần như sau: ([] ~ []).

a) : Val = 2 + 1 = 3 → Val = 2 + 0 = 2 → Val = 2 + 1 = **3**
b) : Val = 0 + 1 = 1→ Val = 1 + 0 = 1 → Val = 1 + 1 = **2**
c) : Val = 0 × 2 + 1 = 1 → Val = 1 × 2 + 0 = 2 → Val = 2 × 2 + 1 = **5**
d) : Val = 0 / 2 + 1 = 1 → Val = 1 / 2 + 0 = 0.5 → Val = 0.5 / 2 + 1 = **1.25**

Khi bạn biết giá trị của kết quả đúng, như trong trường hợp này, thử mỗi đáp án để tìm ra kết quả là một phương pháp hiệu quả. Vì việc thay vào $1\times2^2 + 0\times2^1 + 1\times2^0 = ((1\times2) + 0)\times2+1$ là một cách hiệu quả để giảm khối lượng tính toán, và nó xuất hiện trong cả bài kiểm tra buổi sáng lẫn buổi chiều.

Các ô trống B - D:

Nhớ lại là hàm `IntToN` dùng để chuyển một số nguyên thành chuỗi số trong hệ N. Hàm này dài hơn một chút so với `MToInt`, và do đó có nhiều ô trống hơn, nhưng đừng lo lắng về điều đó. Hãy tự tin vào bản thân. Đầu tiên, chú ý giá trị trả về là chuỗi số đã chuyển, nội dung của nó thấy trong biến `Tmp`. Xét về mặt cấu trúc, quá trình lặp là phần chính, và chúng ta nhìn thấy là chương trình nối các ký tự (số) trong biến `Tmp` trong khi số dư thu được ở mỗi bước. Giữ điều điều đó trong đầu, nhớ cách thực hiện chuyển cơ số. Ví dụ, cách để chuyển giá trị số 5 sang hệ nhị phân, số được lặp đi lặp lại việc chia cho 2 trong khi lưu trữ số dư tại mỗi lần chia.



Hình 1

Không nên chỉ nhìn vào chương trình viết bằng giả ngôn ngữ. Nó rất quan trọng khi bạn nhớ các ví dụ như ví dụ trên.

Với ô trống B, bạn có thể tìm ra đó có thể là sự khởi tạo biến, xem xét vị trí tương đối của nó với câu sau (`Tmp ← ""`). Trong khởi tạo biến, điều quan trọng là xác định biến cần được khởi tạo cho mục sau đó. Điều kiện của vòng lặp là `"Quo >= Rdx"`, và `Rdx` là một tham số, do đó giá trị của nó phải có rồi. Nhưng `"Quo"` (thương số) là một biến được định nghĩa trong hàm này, vì thế khởi tạo giá trị của nó trước khi so sánh là cần thiết. Vậy đây là câu lệnh khởi tạo cho biến `Quo`.

Tiếp đó, xem xét biến `Quo`. Tại thời điểm ban đầu của vòng lặp, chúng ta thấy `"Rem ← Quo % Rdx"`; số dư thu được khi chia `Quo` cho `Rdx` chứa trong biến `Rem`. (`"%"` là toán tử Môđun, nhưng bạn có thể tìm ra nếu bạn không biết điều đó bởi vì tên `Rem` có nghĩa là số dư). Chú ý điều này và ví dụ trong hình 1, bạn có thể thấy đây có thể là một phần trong việc thực hiện chuỗi biến đổi 5 → 2. Nếu là trường hợp này, lời chú thích "thương số" cho biến cũng đáng

chú ý. Giá trị khởi tạo của nó là số nguyên trước khi chuyển, do đó đáp án đúng là (e) `Quo ← Val`. Chú ý là tham số `Val`, là số nguyên trước khi chuyển, không được sử dụng ở chỗ nào khác. Tham số này là giá trị cần thiết nhưng không tìm thấy ở đâu trong hàm. Điều đó có nghĩa là giá trị của nó đư truyền cho một biến khác. Đó chính là một dấu hiệu cho bạn chọn đáp án đúng.

Với ô trống C, xem xét nội dung của vòng lặp. Đầu tiên, `Rem ← Quo % Rdx` gán giá trị số dư cho biến `Rem`, kết quả của phép chia cho `Rdx` là "N" trong hệ cơ số N. Câu lệnh tiếp theo `Tmp ← ToStr(Rem) + Tmp` nối số dư vào biến `Tmp` sau khi chuyển thành một số Bây giờ `Tmp` là một biến chứa xâu ký tự số kết quả. Trong lần lặp tiếp theo, giá trị của `Quo` cần được thay đổi nên phần tương ứng với sự thay đổi này phải được thực hiện ở đây. Trong ví dụ ở hình 1, đó là nơi diễn ra "52". Trong vòng lặp tiếp theo, giá trị của `Quo` phải là thương số thu được khi lấy số chia cho `Rdx`, nhưng thương số này bị bỏ qua. Thậm chí nếu bạn không tìm thấy nơi nào mà giá trị thương số này được lưu lại. Do đó, cần thiết để tìm thương số ở đây và thay giá trị vào biến `Quo`. Vì thế câu trả lời đúng là (a) `Quo ← Quo / Rdx`.

Với ô trống D, phần còn lại của vòng lặp phải được thực hiện. Sử dụng ví dụ trên, biến `Tmp` bây giờ chứa xâu ký tự số tương ứng với thủ tục của hình 1, (ví dụ "01") cho tới thời điểm hiện tại. Vì thế kết quả của bước cuối cùng phải được nối vào và đây là thương số không phải là số dư. Mặt khác, điều kiện dừng của vòng lặp chia là thương số (1) nhỏ hơn số chia (2). Điều này cũng phù hợp với điều kiện lặp của hàm này. Do đó, chúng ta thấy rằng tại điểm này cần thiết để chuyển thương số thành một số và nối vào trước trong chuỗi ký tự. Đáp án đúng là (h) `Tmp ← ToStr(Quo) + Tmp`. Ngoài ra còn có dấu hiệu khác cho ô trống này ở cấu trúc chương tình. Điều duy nhất xảy sau ô trống là đưa ra giá trị trả về. Vì thế không quan trọng giá trị biến nào thay đổi đều không ảnh hưởng tới giá trị trả về, trừ biến `Tmp`. Điều này gợi ý là dòng này liên quan tới biến `Tmp`, từ đó giới hạn các lựa chọn cho đáp án đúng còn (h) và (i). Sau đó, bạn phải tìm cách xác định đáp án đúng.

Ô trống E:

Nhớ lại hàm `ToInt` chuyển tham số của nó, một ký tự số, thành một nguyên. Từ các lựa chọn trả lời và tên của biến `Idx`, bạn có thể đã tìm ra một vài bảng tìm kiếm có liên quan. Vì giá trị trả về là `Idx`, bạn nên biết rằng chỉ số (`Idx`) được trả về khi nào mà bảng được tìm kiếm tuần tự và điều kiện tìm kiếm được thỏa mãn. Nhóm đáp án được chia ra bởi điều kiện cho việc tiếp tục vòng lặp: Trong khi tham số `P` nhỏ hơn (a, c) hoặc lớn hơn (b, d). Đầu

tiên, chúng ta quyết định xem điều kiện nào. Bảng "Code" đang tìm kiếm có các ký tự "0" tới "F" theo thự tự tăng dần của mã ký tự. Giá trị khởi tạo cho tìm kiếm là `Idx = 0`, nên quá trình tìm kiếm bắt đầu từ thành phần đầu tiên. Mặt khác, trong quá trình tìm kiếm rất khó khăn để giữ điều kiện P nhỏ hơn. Bây giờ chúng ta biết rằng bất đẳng thức đúng là `>` hoặc `>=`, chúng ta chỉ phải quyết định có = hay không; điều đó cho chúng ta câu trả lời đúng. Ví dụ, khi các tham số là "0", chúng ta có `Idx = 0`. Nhưng khi điều kiện là `P >= Code[Idx]` phép so sánh đầu tiên đã thỏa mãn điều kiện và `Idx` được cộng vào, kết quả là `Idx = 1`. Bởi vậy câu trả lời đúng sẽ không có dấu =, và do đó đáp án đúng là (b) `P > Code[Idx]`.

Tại đầu câu hỏi này, chúng ta nhìn bao quát tất cả các hàm; tuy nhiên, trong bài thi thực tế, bạn nên nhớ lấy các ô trống và theo dòng chương trình khi cần thiết phải thực hiện.

Bạn có thể thấy là câu hỏi này chứa nhiều ví dụ nhắc bạn nhớ đến các tính chất của chương trình trong ngôn ngữ C, ví dụ sử dụng giá trị trả về của hàm mà không thay thế chúng vào các biến. Các thuật ngữ như "`ToInt`" và "`ToStr`" thì rõ ràng như C. Xu hướng này sẽ còn tiếp tục được sử dụng, vì thế mọi người nên có các kiến thức cơ bản về ngôn ngữ C. Tuy nhiên, bạn không bắt buộc biết rõ về cú pháp hoặc có khả năng viết chương trình với ngôn ngữ C.

Câu hỏi 2

Q2. Đọc mô tả chương trình, giải thích cú pháp cho giả ngôn ngữ, và bản thân chương trình dưới đây, sau đó trả lời các câu hỏi con 1 và 2.

[Mô tả chương trình]

Đây là chương trình con QuickSort, chương trình sắp xếp các số nguyên từ A[Min] đến A[Max](0 <= Min < Max) trong mảng một chiều A.

(1) Thủ tục sắp xếp như sau.

(i) Chương trình tìm kiếm tuần tự trong mảng A[Min+1] đến A[Max] một phần tử, có giá trị khác so với giá trị của A[Min], lấy phần tử tìm thấy đầu tiên, so sánh với A[Min], và chọn bất cứ giá trị nào lớn hơn làm giá trị tham chiếu (Pivot). Nếu tất cả các phần tử của mảng là giống nhau, quá trình sắp xếp kết thúc. Chương trình con FindPivot được sử dụng để chọn giá trị tham chiếu.

(ii) Các phần tử được sắp xếp lại sao cho tất cả các phần tử nh pivot là A[Min], ..., A[i-1] (Min < i <= Max) và tất cả các phần tử lớn hơn hoặc bằng giá trị pivot là A[i], ..., A[Max]. Chương trình con Arrange thực hiện quá trình này.

(iii) Việc sắp xếp lại các phần tử của A[Min], ..., A[i-1] và A[i], ..., A[Max]) được xem như là hai mảng mới và sắp xếp bằng cách áp dụng đệ quy QuickSort.

(2) Chi tiết tham số cho các chương trình con được đưa ra trong các bảng dưới đây.

Bảng 1 Tham số hàm `QuickSort`

Tên biến	Đầu vào/Đầu ra	Ý nghĩa
`A`	Vào/Ra	Mảng một chiều cần sắp xếp
`Min`	Vào	Chỉ số phần tử đầu tiên trong phạm vi sắp xếp
`Max`	Vào	Chỉ số phần tử cuối cùng trong phạm vi sắp xếp

Bảng 2 Tham số hàm `FindPivot`

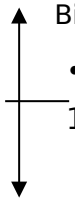
Tên biến	Đầu vào/Đầu ra	Ý nghĩa
`A`	Vào	Mảng một chiều cần sắp xếp
`Min`	Vào	Chỉ số phần tử đầu tiên trong phạm vi sắp xếp
`Max`	Vào	Chỉ số phần tử cuối cùng trong phạm vi sắp xếp
`Ret`	Ra	Trả về chỉ số của phần tử chứa giá trị pivot. Tuy nhiên trả về "`-1`" nếu các phần tử `A [Min]`, ..., `A Max]` có cùng giá trị.

Bảng 3 Tham số hàm `Arrange`

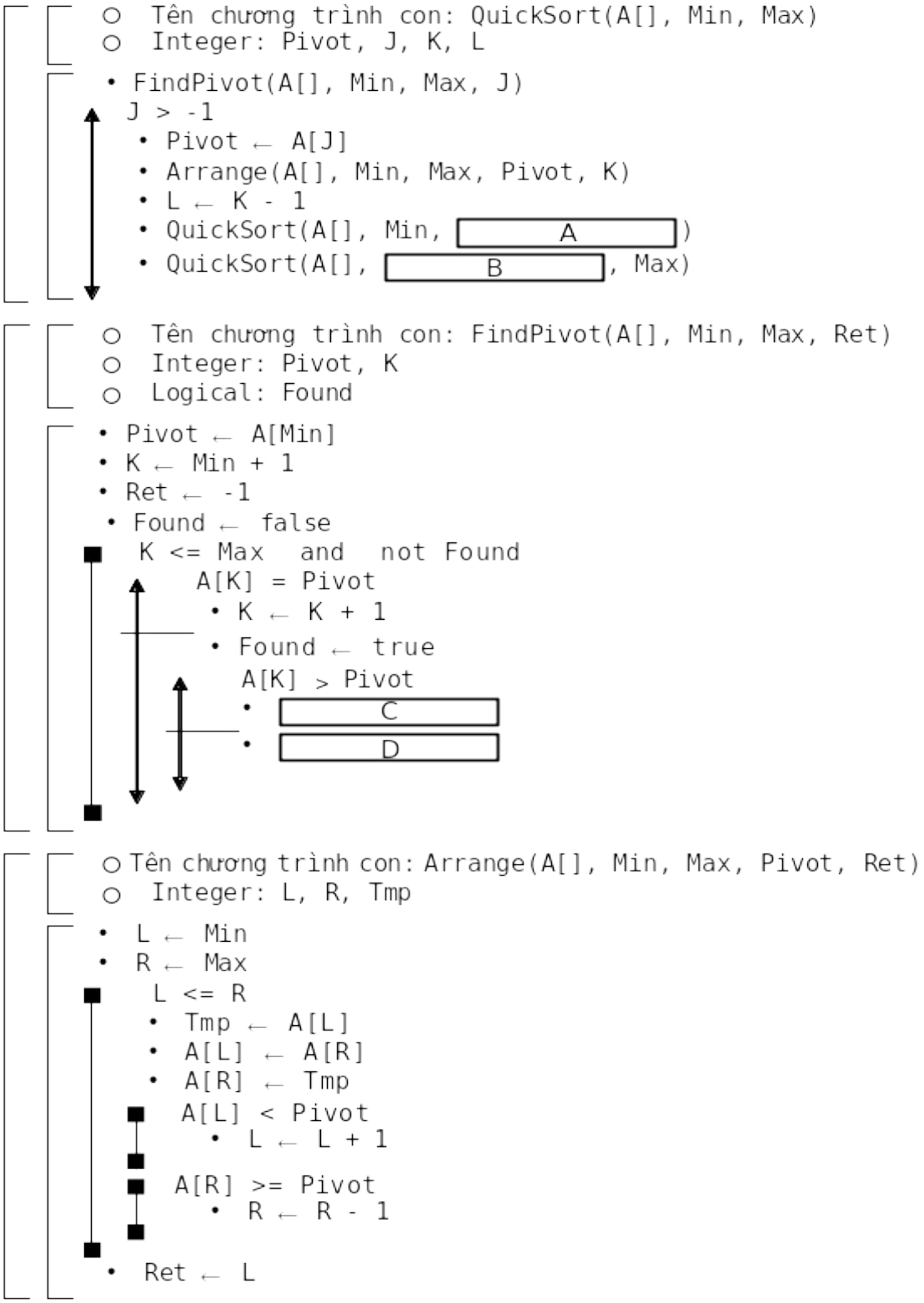
Tên biến	Đầu vào/Đầu ra	Ý nghĩa
`A`	Vào/Ra	Mảng một chiều cần sắp xếp
`Min`	Vào	Chỉ số phần tử đầu tiên trong phạm vi sắp xếp
`Max`	Vào	Chỉ số phần tử cuối cùng trong phạm vi sắp xếp
`Pivot`	Vào	Giá trị tham chiếu
`Ret`	Ra	Sắp xếp lại các phần tử sao cho giá trị của `A[Min]`, ..., `A[i-1]` nhỏ hơn `Pivot` và giá

		trị của A[i] ... A[Max] là lớn hơn hoặc bằng Pivot, và trả về giá trị của i.

[Giải thích cú pháp giả ngôn ngữ]

Cú pháp	Giải thích
	Một vùng liên tiếp nơi khai báo và thực hiện được miêu tả.
	Khai báo tên, kiểu và các thuộc tính khác của thủ tục và biến …
• Biến ← Biểu thức	Gán giá trị của biểu thức vào biến.
{Chú thích}	Ghi chú thích.
	Thể hiện lựa chọn. Khi biểu thức điều kiện là đúng (true), đoạn chương trình 1 được thực hiện, khi là sai (false) thì đoạn chương trình 2 được thực hiện.
	Thể hiện lặp với điều kiện kết thúc ở trên đầu. Khi biểu thức điều kiện là đúng (true) thì đoạn chương trình được thực hiện.

[Chương trình]



Câu hỏi con 1

Từ các nhóm đáp án dưới, hãy chọn đáp án đúng để điền vào các ô trống ☐ trong chương trình ở trên.

Nhóm đáp án cho A và B:

a) K b) L c) Max d) Min

Nhóm đáp án cho C và D:

a) `Ret ← A[K]` b) `Ret ← A[Max]` c) `Ret ← A[Min]`
d) `Ret ← K` e) `Ret ← Max` f) `Ret ← Min`

Câu hỏi con 2

Nội dung của các tham số đã được tổng hợp trong bảng 4 sau khi sử dụng `QuickSort` để sắp xếp các phần tử từ `A[0]` tới `A[9]` của mảng số nguyên một chiều. Từ các nhóm đáp án dưới đây, chọn đáp án đúng để điền vào ô trống ☐ trong bảng sau đây.

Bảng 4 Số lần gọi và nội dung các tham số của hàm `QuickSort`

Số lần gọi	A	Min	Max
vòng đầu tiên	A[0] … A[9]: 3 5 8 4 0 6 9 1 2 7	0	9
vòng thứ 2	3 2 1 4 0 6 9 8 5 7	0	4
vòng thứ 3	E	F	G
⋮	⋮	⋮	⋮
vòng thứ n	0 1 2 3 4 5 6 7 8 9	9	9

Nhóm đáp án cho E:

a) 0 1 2 4 3 6 9 8 5 7

b) 0 2 1 3 4 6 9 8 5 7

c) 0 2 1 4 3 6 9 8 5 7

d) 0 4 1 2 3 6 9 8 5 7

Nhóm đáp án cho F và G:

a) 0 b) 1 c) 2
d) 3 e) 4 f) 5

Đáp án câu 2

Sắp xếp nhanh (Giả ngôn ngữ)

Đáp án đúng

[Câu hỏi con 1] A – b, B – a, C – d, D – f

[Câu hỏi con 2] E – c, F – a, G – c

Giải thích

Sắp xếp nhanh (Quick sort) xuất hiện trong bài kiểm tra buổi sáng, khi đó nó được định nghĩa theo cách sau: "Đây là một cách sắp xếp dữ liệu trong đó toàn bộ tập dữ liệu đầu tiên được chia thành hai nhóm, một nhóm nhỏ hơn giá trị đang xét và một nhóm lớn hơn hoặc bằng giá trị này. Quá trình phân nhóm tiếp tục được áp dụng cho mỗi nhóm cho tới khi toàn bộ dữ liệu được sắp. Phương pháp này hiệu quả khi số lượng phép so sánh và chuyển đổi là nhỏ, tuy nhiên nhiều người vẫn có cảm giác là quá trình đệ quy tự nhiên gây khó khăn cho việc quản lý. Ở nhiều kỳ thi, khi họ nhìn thấy câu hỏi này trong bài thi, họ tự nghĩ, "Ôi không! Tôi biết rằng tôi nên học thuật toán quicksort", mặc dù các thí sinh bị bẫy bởi các câu hỏi loại này khi họ suy nghĩ theo cách như vậy. Nhưng các câu hỏi thi được thiết kế với ý định để cho thí sinh dự thi với các kiến thức cơ bản thông thường có thể trả lời, vì thế bạn cần giữ được bình tĩnh khi trả lời các câu hỏi này.

[Câu hỏi con 1]

Đây là câu hỏi điền vào chỗ trống, nhưng bất cứ khi nào một thủ tục cụ thể được miêu tả trong phần [Mô tả chương trình] giống như câu hỏi này, tốt nhất là đầu tiên nhìn tổng quát chương trình và xem xét câu hỏi [Mô tả chương trình] phù hợp với chương trình. Giống như chủ đề câu hỏi "Quick Sort," bất cứ khi nào một câu hỏi liên quan tới một thuật toán mà không ai có thể dễ dàng giải quyết, câu hỏi có xu hướng xuất hiện theo cách này.
Dựa trên [Mô tả chương trình], thủ tục sắp xếp bởi chương trình con `QuickSort` là như sau:

(i) Sử dụng chương trình con `Partition` để tìm ra giá trị tham chiếu (`Pivot`).
(ii) Sử dụng chương trình con `Arrange` để tái sắp xếp các phần tử trong mảng

(A[Min],…, A[i - 1] nhỏ hơn Pivot và A[i],…, A[Max] lớn hơn hoặc bằng giá trị Pivot).

(iii) Xem xét A[Min],…, A[i - 1] và A[i],…, A[Max] như là các mảng mới và áp dụng đệ quy thủ tục QuickSort.

Đầu tiên, chúng ta hãy chú ý tới tên của các chương trình con sử dụng trong mục này và xem xét sự tương ứng của chúng với c QuickSort; không khó để xác định như sau.

```
○ QuickSort(A[], Min, Max)
○ Integer: Pivot, J, K, L
□ FindPivot(A[], Min, Max, J)                    ..... (i)
▲ J > –1
  □ Pivot ← A[J]
  □ Arrange(A[], Min, Max, Pivot, K)             ..... (ii)
  □ L ← K - 1
  □ QuickSort(A[], Min,      A     )             } ..... (iii)
  □ QuickSort(A[],      B     , Max)             }
▼
```

Ô trống A và B:

Chúng ta tiếp tục phần giải thích với giả sử là chúng ta biết lời gọi đệ quy thủ tục `QuickSort` thứa 2 ô trống này, tương ứng với (iii). Bảng 1 đưa ra danh sách các tham số của `QuickSort`; có 3 tham số là theo thứ tự "`A` (mảng để sắp xếp), `Min` (chỉ số của phần tử đầu tiên trong phạm vi sắp xếp), và `Max` (chỉ số của phần tử cuối cùng trong phạm vi sắp xếp"). Xem xét cái để chèn vào các ô trống này, đưa ra nội dung các tham số này. Ô trống A phải là chỉ số của phần tử cuối cùng trong phạm vi sắp xếp, tương ứng với "`i - 1`" trong (iii) của mô tả. Ô trống B là chỉ số của phần tử đầu tiên trong phạm vi sắp xếp, tương ứng với "`i`" trong (iii). Bây giờ, chú ý hai giá trị, "`i - 1`" và "`i`", chúng ta đọc trong (ii) của mô tả, "tất cả các phần tử nhỏ hơn `Pivot` là `A[Min]`, ..., `A[i - 1]` và tất cả các phần tử bằng hoặc lớn hơn `Pivot` là `A[i]`, ..., `A[Max]`". Vì thế, "`i - 1`" là chỉ số của phần tử cuối cùng nhỏ hơn `Pivot` trong khi "`i`" là chỉ số của phần tử đầu tiên lớn hơn hoặc bằng `Pivot`. Các phần tử của mảng để sắp xếp `A[Min]`, ..., `A[Max]`, được đưa ra theo thứ tự này tại thời điểm bắt đầu. Bằng `FindPivot` của (i), một giá trị `Pivot` được chọn, và bằng `Arrange` trong (ii), các phần tử được sắp xếp. Nhìn vào các tham số của hàm `Arrange` trong bảng 3. Có 5 tham số, nhưng tham số cuối cùng `Ret` nói rằng "Tái sắp xếp các phần tử sao cho giá trị `A[Min]`, ..., `A[i - 1]` là nhỏ hơn `Pivot` và giá trị `A[i]`, ..., `A[Max]` là lớn hơn hoặc bằng `Pivot`." Trong chương trình, `K` được sử dụng thay cho `Ret` (chú ý "`Arrange(A[],Min,Max,Pivot,K)`"), do đó giá trị `K` này tương ứng với "`i`" trong (iii). Do đó ô trống A có thể là "`K - 1`", nhưng trong nhóm trả lời không có lựa chọn này. Nhìn trở lại chương trình, chúng ta thấy rằng có một dòng `L ← K – 1` ngay sau khi gọi `Arrange`. Vì vậy `L = K – 1`, và đáp án đúng cho ô trống A là (b). Bây giờ đó là dấu hiệu cho thấy đáp án đúng cho ô trống B phải là (a) `K`.

Ô trống C và D:

Các ô trống này nằm trong chương trình con `FindPivot`, đã giải thích trong

[Mô tả chương trình](i). Mô tả (i) hơi dài, nhưng nội dung của nó có thể được tóm tắt như sau: " Tìm kiếm tuần tự trong mảng từ A[Min+1] đến A[Max] một phần tử có giá trị khác với A[Min] (Bao gồm tất cả các giá trị ngoại trừ A[Min]), lấy phần tử đầu tiên tìm thấy, so sánh với A[Min], và chọn bất cứ giá trị nào lớn hơn làm giá trị tham chiếu (Pivot) ". Bây giờ xem xét các nhóm trả lời. Chú ý là tất cả các lựa chọn đều thiết lập Ret. Quan tâm tới Ret, từ bảng 2 giải thích các tham số của FindPivot "Trả về chỉ số của phần tử chứa giá trị tham chiếu". Chúng ta thấy rằng các ô trống này dùng để quyết định giá trị tham chiếu và thiết lập chỉ số của tham số Ret. Phần này chứa những ô trống được thực hiện khi điều kiện "A[K] = Pivot" là sai. Trong lúc đó, tại thời điểm đầu của chương trình, chúng ta đọc thấy "Pivot ← A[Min]" dẫn tới tại thời điểm này "Pivot = A[Min]". Vì vậy, điều kiện "A[K] = Pivot" là sai nghĩa là giá trị của A[K] khác với giá trị của A[Min]. Đúng hoặc Sai của điều kiện "A[K] > Pivot" quyết định giá trị của tham số Ret, phần này chính xác tương ứng với phần "chọn Pivot" đã mô tả ở trên. Ô trống C áp dụng khi A[K] > Pivot (= A[Min]). Nhớ lại Pivot là giá trị lớn hơn gặp (khác với A[Min]) và A[Min], và vì A[K] là lớn hơn nên A[K] sẽ trở thành Pivot. Ô trống C sau đó (d) Ret ← K. Ngược lại, ô trống D là trường hợp khác vì thế đáp án đúng là (f) Ret ← Min. Các đáp án từ (a) tới (c) đều thiết lập một phần tử của mảng cho tham số Ret. Bằng cách đọc cẩn thận giả thích trong bảng 2 ("trả về chỉ số phần tử chứa giá trị tham chiếu"), khi đó các đáp án sai ngay lập tức có thể được loại bỏ. Bởi vậy bạn còn lại có 3 lựa chọn. Sắp xếp nhanh nghe có vẻ là một khái niệm khó hiểu, nhưng hi vọng các bạn tin rằng đó không phải là một câu hỏi khó.

[Câu hỏi con 2]

Đây là câu hỏi truy vết. Các ô trống trong chương trình đã được điền vào, vì thế cơ bản là không có lựa chọn nhưng để chèn vào nội dung của các ô trống và theo vết chương trình một cách cẩn thận. Đây là câu hỏi con để kiểm tra bạn với trọng tâm của câu hỏi, đó là sắp xếp nhanh, thủ tục đệ quy.

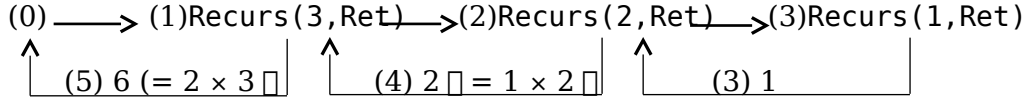
Đầu tiên chúng ta xem lại về đệ quy. Đệ quy có nghĩa là “Gọi nó”. Một ví dụ thường được sử dụng để giả thích về đệ quy là hàm tính giai thừa. Chương trình được viết như sau. Nhớ lại là giai thừa của một số là kết quả phép nhân các số từ 1 cho tới số đó. Ví dụ giai thừa của3 (3!) là 1 × 2 × 3 = 6.

[Ví dụ]

```
○ Recurs(n, Ret)
○ integer: n, Ret
  n > 1
   Recurs(n-1, Ret)
   Ret ← n × Ret
  ──────────────
   Ret ← 1
```

(Trong giả ngôn ngữ sử dụng với các câu hỏi thi, các tự hàm dường như không thể trả về giá trị, vì thế có vẻ khác so với chương trình thực tế viết trên ngôn ngữ giống như C).

Bây giờ, xem xét lời gọi "Recurs(3,Ret)" thực hiện thế nào khi sử dụng chương trình trên. Đầu tiên, vì tham số truyền vào là 3 nên giá trị của tham số n là “3”, thỏa mãn "3 > 1", tiếp đó "Recurs(2,Ret)" được gọi. Tham số n là 2, và bởi vì "2 > 1", sẽ gọi hàm "Recurs(1,Ret)". Tiếp theo, khi tham số n là 1, và vì "1 > 1" là sai, nên tham số Ret được thiết lập là 1, và chương trình quay về nơi lời gọi thực hiện. Tại lời gọi "Recurs(2,Ret)", giá trị của Ret (= 1) được sử dụng để thực hiện "Ret ← Ret (= 1) × n (= 2)". Sau đó nó lại quay về thời điểm thực hiện lời gọi (Recurs(3,Ret)), với "2" là giá trị của Ret. Điều đó làm cho chương trình hoàn "Recurs(3,Ret)" khi nó thực hiện "Ret ← Ret (= 2) × n (= 3)". Kết quả Ret là 6, là kết quả đúng. Biểu đồ sau thể hiện các bước thực hiện chương trình để thu kết quả.



Như nhìn thấy ở đây, chương trình gọi chính nó, nhưng theo thứ tự để có thể kết thúc quá trình, hàm được gọi cuối cùng phải kết thúc.

Bây giờ, chúng ta xem xét chương trình được cho trong câu hỏi. Như bảng 4 gợi ý, QuickSort(A[],0,9) được gọi đầu tiên. Tiếp đó, thực hiện lời gọi thứ hai QuickSort(A[],0,4). Tại thời điểm này thấy rằng lời gọi đầu tiên chia mảng A thành A[0], ..., A[4], có giá trị nhỏ hơn giá trị Pivot, và A[5], ..., A[9], có giá trị lớn hơn hoặc bằng giá trị Pivot. Vì thế, theo các nhóm này (sắp xếp bởi Arrange), trên bề mặt là gọi các hàm "QuickSort(A[],0,4)" và "QuickSort(A[],5,9)". Tuy nhiên, chú ý các lời gọi không cần thiết theo đúng thứ tự này. Như đã đề cập trước, trước tiên bên trong của việc thực hiện lời gọi hàm QuickSort(A[],0,4) một lời gọi đệ quy khác được thực hiện, đây là lời gọi được thực hiện đầu tiên.

```
QuickSort(A[],0,9)      ← Lời gọi đầu tiên
   ...
 FindPivot và Arrange chia mảng thành hai nhóm: A[0],..., A[4] và A[5],..., A[9].
   ...
 QuickSort(A[],0,4)     ← Lời gọi thứ hai.
 QuickSort(A[],5,9)     ← Có thể không là lời gọi thứ ba.
                Lời gọi trong khi QuickSort(A[],0,4) được thực hiện trước.
```

Chúng ta theo vết của lời gọi thứ hai (QuickSort(A[],0,4)). Phạm vi các đối tượng để sắp xếp là A[0], ..., A[4], và giá trị của các phần tử như sau (Lấy từ bảng 4).

0	1	2	3	4
3	2	1	4	0

Trước tiên, xem xét Pivot, là giá trị thu được bởi hàm FindPivot. Giá trị khác đầu tiên từ A[0] (=A[Min]) =3, ví dụ, giá trị lớn hơn giữa A[1]=2 và A[0]=3, vì thế A[0] = 3 được chọn. Tiếp đó, thiết lập Pivot= 3, bây giờ chúng ta theo vết hàm Arrange(A[],0,4,3,K).

Bắt đầu chương trình có "L ← Min " và "R ← Max" vì thế L = 0 và R = 4. Sau đó, chương trình vào vòng lặp với điều kiện thỏa mãn L <= R.

(i) Đầu tiên có 3 câu lệnh "Tmp ← A[L]", "A[L]← A[R]", và "A[R]← Tmp". Đây là cách chuẩn để chuyển đổi hai giá trị của hai biến. Nói cách khác giá trị của A[0] và A[4] được đổi cho nhau, dẫn tới A[0] = 0 và A[4] =

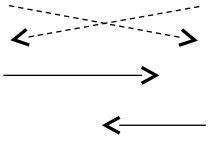
3.

(ii) Trong vòng lặp tiếp theo, khi `A[L] < Pivot`, giá trị của `L` được tăng lên 1 mỗi lần. Vì `A[0] = 0, A[1] = 2, A[2] = 1, A[3] = 4`, ta thấy là `A[3] < Pivot` không thỏa mãn nên kết quả là `L = 3`.

(iii) Tiếp theo, trong vòng lặp dưới đây, trong khi "`A[R] >= Pivot`", giá trị của `R` được giảm đi 1 mỗi lần. Vì `A[4] = 3, A[3] = 4, và A[2] = 1,` ta thấy là `A[2] >= Pivot` không thỏa mãn nên kết quả là `R = 2`.

(iv) Vì `L = 3` và `R = 2`, điều kiện lặp không thỏa mãn nên chương trình con được hoàn thành. Biểu đồ dưới đây thể hiện quá trình.

01234PivotLR321403(i)0403(ii)0→30214(iii)4→2
143(iv)3202143

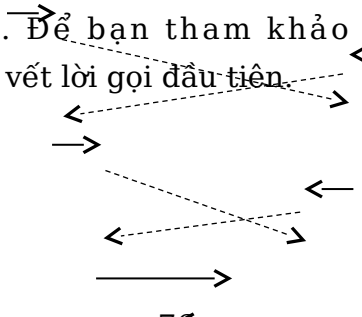


Khi vòng lặp kết thúc, giá trị L (= 3) tại thời điểm đó sẽ được gán cho Ret, và chương trình quay về nơi lời gọi đầu tiên thực hiện.

Bây giờ chúng ta xem xét các ô trống. Đầu tiên nhìn và A[0] tới A[4] là trong điều kiện (iv) của biểu đồ trên. A[5] tới A[9] không thay đổi, vì thế chúng ta kế thừa điều kiện từ lời gọi thứ hai (trong tất cả các lựa chọn của nhóm trả lời, phần này là xác định). Vì thế đáp án đúng là (c). Tiếp theo xem xét ô trống F và G. Bằng cách truy vết thấy rõ ràng là giá trị của Ret (= K) là 3 và vì thế A[0], …, A[2] là các giá trị nhỏ hơn giá trị Pivot trong khi A[3] và A[4] là các giá trị lớn hơn hoặc bằng gí trị Pivot. Với kết quả này, lời gọi thứ hai (QuickSort(A[], 0, 4)) được thực hiện, trong đó các lời gọi sâu hơn "QuickSort(A[], 0, 2)" và "QuickSort(A[], 3, 4)" sẽ được thực hiện. Vì thế lời gọi thứ 3 là "QuickSort(A[], 0, 2)", đáp án đúng cho ô trống F là (a) 0, và đáp án cho ô trống G là (c) 2.

Bây giờ, bạn nên học khả năng tìm ra cách để tìm được đáp án. Nếu bạn đã học về thuật toán sắp xếp nhanh trước khi kiểm tra, bạn sẽ cảm thấy dễ dàng và tự tin, và có lẽ bạn sẽ có nhiều thuận lợi. Tuy nhiên nếu bạn chưa học, câu hỏi này không phải là câu hỏi không thể trả lời miễn là bạn biết về chủ đề này với (quá trình lặp đệ quy thủ tục chia nhóm, trong đó dữ liệu được phân chia thành hai nhóm, một nhóm có giá trị lớn hơn và nhóm còn lại có giá trị nhỏ hơn hoặc bằng với một giá trị cho trước) và hiểu lời gọi ban đầu của quá trình đệ quy. Các câu hỏi thuật toán trong bài thi FE thông thường luôn chú trọng tới khả năng để truy vết hơn là khả năng xây dựng thuật toán; và trong vài năm gần đây, khả năng đọc hiểu chương trình càng trở nên quan trọng. Trong phần giải thích này chúng ta truy vết lời gọi thứ hai nhưng tốt nhất là nên truy vết lời gọi đầu tiên cũng như là lời gọi thứ ba và các lời gọi sau đó. Để bạn tham khảo chúng tôi kết thúc mục này bằng cách đưa ra biểu đồ truy vết lời gọi đầu tiên.

0123456789PivotLR3584069127Giải thích50973Đảo vị trí07Bước lặp với L93Bước lặp với R0937Đảo vị trí0→135Bước lặp với L9→827Bước lặp với R1825Đảo vị trí1→228Bước lặp với L8→735Bước lặp với R2718Đảo vị trí2→514Bước lặp với L7→40698Bước lặp với R54321406985714→R



Thiết kế chương trình

7

[Phạm vi các câu hỏi]
Qui trình phát triển hệ thống,
Quá trình thiết kế chương trình, thiết kế có cấu trúc,
Thiết kế module,
Tài liệu thiết kế chương trình,....

Câu hỏi 1

Q1. Đọc mô tả sau về thiết kế chương trình và trả lời các câu hỏi con từ 1 đến 4.

Một hệ thống được thiết kế để cung cấp những tin tức trong nước dựa vào sở thích của người dùng. Dưới đây là phác thảo về hệ thống

[Phác thảo hệ thống]

(1) Tin tức cần cung cấp được lưu trong một tệp tin tức theo định dạng dưới đây. Tệp tin tức này là một tệp tuần tự bao gồm các bản ghi, mỗi bản ghi chứa một mục tin tức, và các mục tin tức được ghi lại theo thứ tự mà chúng được đăng kí.

Định dạng bản ghi của tệp tin tức

Ngày đăng Kí	Thời gian đăng kí	Ngày xảy ra	Thời gian xảy ra	Loại	Tiêu đề	Tóm tắt	Chi tiết	Địa chỉ tệp ảnh

Ngày đăng kí là một chuỗi kí tự có 8 chữ số đại diện cho năm, tháng và ngày khi mục tin tức được đăng kí. Ví dụ “20080401” biểu diễn cho ngày mùng 1 tháng 4 năm 2008. Thời gian đăng kí là chuỗi kí tự có 4 chữ số đại diện cho giờ và phút khi mục tin tức được đăng kí. Ví dụ :”1830” thay thế cho 6h30 chiều. Ngày xảy ra là năm, tháng và ngày xảy ra sự kiện của tin tức. Ngày xảy ra có cùng định dạng với ngày đăng kí. Thời gian xảy ra là thời điểm xảy ra sự kiện nêu trong tin tức và có cùng định dạng với thời gian đăng kí. Loại là loại tin tức, và lưu trữ một trong các loại sau: “Chăm sóc sức khỏe”, “Giáo dục”, “Kinh tế”, “Giải trí”, “Khoa học”, “Xã hội”, “Thể thao”, và “Chính trị”. Bên cạnh đó, cũng cần lưu thông tin về dòng tiêu đề, tóm tắt của mục tin, nội dung chi tiết và địa chỉ tệp ảnh liên quan.

(2) Thông tin người dùng được đăng kí trong tệp người dùng theo định dạng sau. Tệp người dùng là một tệp được đánh chỉ mục sử dụng khóa là định danh người dùng (user ID).

Định dạng bản ghi của tệp người dùng

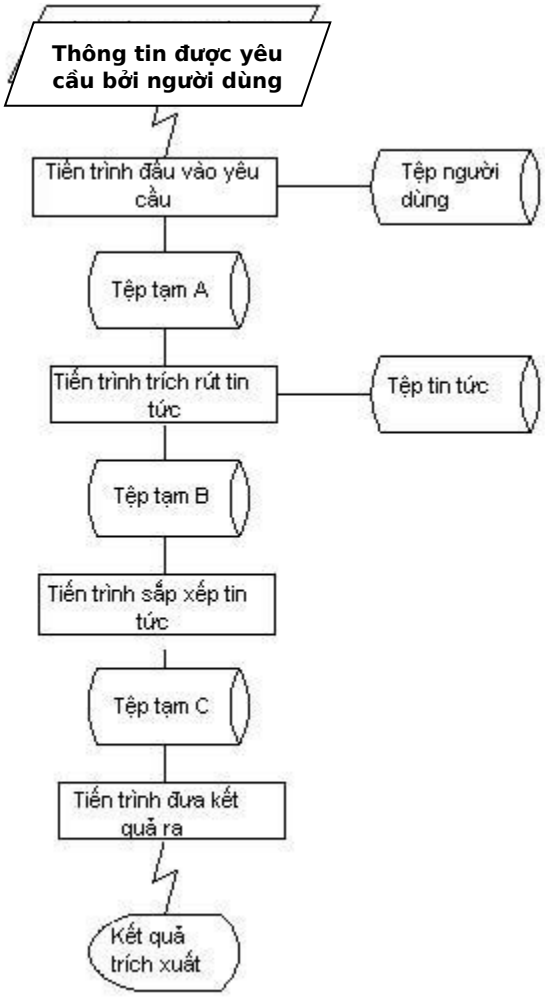
Định danh người dùng (User ID)	Loại 1	Loại 2	Loại 3	Loại 4	Loại 5	Ngày sử dụng cuối cùng	Thời gian sử dụng cuối cùng

Tệp người dùng đăng kí tối đa 5 loại cho các tin tức ưa thích của người dùng trong các trường từ loại 1 đến loại 5. Nếu ít hơn 5 loại thì trường loại còn lại sẽ lưu giá trị NIL (rỗng). Hệ thống cung cấp tối đa 10 mục tin tức lùi dần theo ngày giờ cho mỗi loại đã đăng kí.

Ngày sử dụng cuối cùng và thời gian sử dụng cuối cùng là ngày và thời điểm khi người dùng sử dụng hệ thống lần cuối. Ngày sử dụng cuối cùng và thời gian sử dụng cuối cùng là chuỗi kí tự có định dạng lần lượt giống với ngày đăng kí và thời gian đăng kí.

(3) Nếu quá 24 giờ đã trôi qua từ ngày giờ của lần sử dụng cuối cùng, tất cả mọi mục tin tức đã đăng kí trong 24 giờ trước được đưa ra để trích rút. Nếu lần sử dụng hệ thống trước đó của một người dùng trong vòng 24 giờ, thì chỉ những mục tin tức đã được đăng kí sau thời điểm sử dụng cuối cùng được đưa ra để trích rút.

(4) Dựa vào loại thiết bị cuối của người dùng, hệ thống sẽ thay đổi định dạng của kết quả trích rút cần cung cấp. Nếu thiết bị cuối của người dùng là điện thoại di động, thì kiểu thiết bị cuối thích hợp là “Đơn giản”, và hệ thống đưa ra một đoạn tóm tắt của của mỗi mục tin tức. Nếu thiết bị cuối của người dùng là một máy tính cá nhân thì một kiểu thiết bị cuối tương thích là “Chi tiết”, và hệ thống sẽ cung cấp những chi tiết của mỗi tin tức và các hình ảnh liên quan.


Thông tin được yêu cầu bởi người dùng
Tiến trình đầu vào yêu cầu
Tệp người dùng
Tệp tạm A
Tiến trình trích rút tin tức
Tệp tin tức
Tệp tạm B
Tiến trình sắp xếp tin tức
Tệp tạm C
Tiến trình đưa kết quả ra
Kết quả trích xuất

Hình. Luồng tiến trình

[Mô tả tiến trình]

(1) Trong tiến trình nhập yêu cầu, một bản ghi có định danh người dù giống với định danh người dùng có trong thông tin được yêu cầu sẽ được chọn từ trong tệp người dùng. Thông tin được yêu cầu từ người dùng có định dạng sau.
Loại liên quan, "Đơn giản" hay "Chi tiết" được lưu trong trường thiết bị cuối.

Định dạng của thông tin được yêu cầu

Định danh người dùng	Kiểu thiết bị cuối

Các bản ghi trong định dạng sau được ghi vào tệp tạm *A* từ thông tin được yêu cầu và bản ghi được lựa chọn.

Định dạng bản ghi của tệp tạm *A*

Loại 1	Loại 2	Loại 3	Loại 4	Loại 5	Ngày bắt đầu trích rút	Thời gian bắt đầu trích rút	Kiểu thiết bị cuối

Cuối cùng, ngày của lần sử dụng cuối và thời gian của lần sử dụng cuối trong bản ghi người dùng liên quan trong tệp người dùng được cập nhật theo ngày giờ hiện tại.

(2) Trong tiến trình trích rút tin tức, dựa vào tệp tạm *A*, bản ghi thỏa mãn 2 điều kiện sau được trích rút đồng thời từ tệp tin tức.
(i) Ngày giờ đăng kí của bản ghi tin tức trong tệp tin tức gần hiện tại hơn so với ngày giờ bắt đầu trích rút từ tệp tạm *A*.
(ii) Loại của các bản ghi tệp tin tức phù hợp với 1 trong các trường từ loại 1 đến loại 5 trong tệp tạm *A*. Thông tin cần thiết được thêm vào bản ghi thỏa mãn những điều kiện này và sau đó bản ghi này được ghi vào tệp tạm *B*.

(3) Trong tiến trình sắp xếp tin tức, tệp tạm B được sắp xếp rồi ghi vào tệp tạm *C*.

(4) Trong tiến trình đưa kết quả ra, tối đa 10 mục được trích rút cho mỗi loại từ tệp tạm *C*, và sau đó kết quả được ghi theo định dạng sau để phù

hợp với kiểu thiết bị cuối.

Định dạng được sử dụng khi kiểu thiết bị cuối là “Đơn giản”:

Ngày xảy ra	Thời gian xảy ra	Loại	TIêu đề	Tóm tắt

Định dạng được sử dụng khi kiểu thiết bị cuối là “Chi tiết”:

Ngày xảy ra	Thời gian xảy ra	Loại	TIêu đề	Tóm tắt	Địa chỉ tệp ảnh

(5) Ở đây, thông tin giữa các tiến trình được chuyển cho nhau chỉ khi dùng các tệp được chỉ ra trong hình vẽ.

Câu hỏi con 1

Từ nhóm các câu trả lời dưới đây, hãy chọn câu trả lời đúng để điền vào ô trống [ ] trong mô tả sau.

Một bản ghi của người dùng trong tệp người dùng được chỉ ra dưới đây:

XR205	Kinh tế	Giải trí	NIL	NIL	NIL	20080401	2038

Nếu người dùng này sử dụng hệ thống lúc 17:00 ngày 3 th 2008, [A] được lưu trong trường Thời gian bắt đầu trích rút trong tệp tạm *A* bởi tiến trình nhập yêu cầu và [B] được lưu trong trường Thời gian bắt đầu trích rút.

Nhóm câu trả lời:

a) "0000" b) "1700" c) "2038"
d) "20080401" e) "20080402" f) "20080403"

Câu hỏi con 2

Định dạng bản ghi của tệp tạm *B* được ghi bởi tiến trình trích rút tin tức được chỉ ra dưới đây. Từ nhóm các câu trả lời ở dưới, chọn câu trả lời đúng để điền vào các ô trống [ ] trong định dạng này.

Ngày xảy Ra	Thời gian xảy ra	[C]	Tiêu đề	Tóm tắt	Chi tiết	Địa chỉ của tệp ảnh	[D]

Nhóm câu trả lời:

a) Kiểu thiết bị cuối
b) Thời gian bắt đầu trích rút
c) Ngày bắt đầu trích rút
d) Thời gian đăng kí
e) Ngày đăng kí
f) Loại
g) Định danh người dùng

Câu hỏi con 3

Từ nhóm câu trả lời ở dưới, chọn câu trả lời đúng để điền vào các ô trống [ ] trong mô tả sau.

Trong tiến trình sắp xếp tin tức, các bản ghi thu được trong tiến trình trích rút tin tức được đưa ra để sắp xếp. Trong trường hợp này, khóa sắp xếp đầu tiên là [E], và các bản ghi được sắp xếp theo thứ tự tăng các mã kí tự. Khóa sắp xếp thứ hai là [F], và các bản ghi được sắp xếp thep thứ tự giảm dần. Khóa sắp xếp thứ 3 là [G], và các bản ghi được sắp xếp theo thứ tự giảm dần.

Nhóm câu trả lời:

a) Tóm tắt
b) Địa chỉ tệp ảnh
c) Chi tiết
d) Thời gian bắt đầu trích rút
e) Thời gian đăng kí
f) Ngày đăng kí
g) Thời gian xảy ra
h) Ngày xảy ra
i) Loại

Câu hỏi con 4

Từ nhóm câu trả lời ở dưới, chọn câu trả lời đúng để điền vào các ô trống [ ] trong mô tả sau.

Một bản ghi của người dùng trong tệp người dùng được chỉ ra dưới đây:

DT512	Chính trị	Kinh tế	Thể thao	NIL	NIL	20080401	0400

Số mục tin tức được đăng kí cho mỗi loại lúc 11:00 ngày 1 tháng 4 năm 2008 được chỉ ra dưới đây. Nếu người dùng đó sử dụng hệ thống tại cùng thời điểm trong cùng ngày đó, số lượng bản ghi được ghi vào tệp tạm thời bởi tiến trình sắp xếp tin tức là [H], và số mục tin tức được hiển thị bởi tiến trình đưa kết quả ra là [I].

Bảng Số mục tin tức được đăng kí

Loại	Số mục tin tức được đăng kí mỗi giờ

	00:00 đến 00:59	01:00 đến 01:59	02:00 đến 02:59	03:00 đến 03:59	04:00 đến 04:59	05:00 đến 05:59	06:00 đến 06:59	07:00 đến 07:59	08:00 đến 08:59	09:00 đến 09:59	10:00 đến 10:59
Chăm sóc sức khỏe	0	0	0	0	0	0	2	0	1	0	0
Giáo dục	0	0	0	0	0	1	0	1	0	0	0
Kinh tế	0	0	0	1	0	2	2	6	8	2	4
Giải trí	1	0	1	0	1	2	0	1	2	1	1
Khoa học	1	0	1	0	1	1	3	2	3	2	5
Xã hội	0	1	2	0	1	0	2	3	5	4	3
Thể thao	0	0	0	1	0	0	1	1	0	3	2
Chính trị	0	1	0	0	0	1	4	2	3	6	3

Nhóm câu trả lời:

a) 27 b) 28 c) 50 d) 53

e) 60 f) 98 g) 108

Đáp án câu 1

Cung cấp những tin tức trong nước dựa vào sở thích của người dùng

Đáp án đúng

[Câu hỏi con 1] A – e, B – b

[Câu hỏi con 2] C – f, D – a (thứ tự không liên quan trong C và D)

[Câu hỏi con 3] E – i, F – h, G – g

[Câu hỏi con 4] H – c, I – a

Giải thích

Chủ đề của câu hỏi này là thiết kế chương trình của một hệ thống cung cấp tin tức trong nước dựa vào sở thích của người dùng. Internet được sử dụng rất phổ biến trong những năm gần đây tới mức mà hầu hết các thí sinh đều hình dung được, nhờ vào việc đọc các mô tả của câu hỏi này, một hệ thống trong đó người dùng có thể đăng kí các loại mà người đó quan tâm sao cho những bài báo tin tức đó được hiển thị, từ những bài gần đây nhất, chỉ trong các loại mà người dùng đã đăng kí. Giữ lại khái niệm này trong đầu khi bạn đọc nội dung câu hỏi.

Nội dung câu hỏi gồm có [Phác thảo hệ thống], Luồng Tiến trình (sơ đồ), [Mô tả Tiến Trình], và các câu hỏi con từ 1 đến 4. Về cơ bản, bạn nên đọc các phần trên theo trình tự từ đầu, nhưng bạn cần sáng tạo một chút trong việc phân phối thời gian đọc các câu hỏi. Trong một bài thi bình thường, "Đọc câu hỏi cẩn thận, thiết lập tham số và các điều kiện khác gọn gàng, và sau đó trả lời câu hỏi" là một cách giải quyết tốt; tuy nhiên, trong bài thi thường, đặc biệt là bài thi chiều, thí sinh thường thiếu thời gian. Bạn cần nhớ điều này và bắt kịp với cách đọc các câu hỏi một cách hiệu quả. Thực ra, bạn đọc mô tả của câu hỏi là để trả lời các câu hỏi con. Thậm chí nếu bạn dùng nhiều thời gian đọc mô tả câu hỏi trước câu hỏi con, khi bạn trả lời câu hỏi con, bạn sẽ đọc lại mô tả câu hỏi một lần nữa để chắc chắn. Với điều đó trong ý nghĩ, bạn sẽ đọc câu hỏi với tâm niệm rằng cuối cùng bạn vẫn đọc lại nó một lần nữa.

Đọc [Phác thảo hệ thống] gần như để biết 4 điều sau:

(1) mô tả của tệp tin tức
(2) mô tả của tệp người dùng
(3) mô tả liên quan đến thời gian người dùng sử dụng dịch vụ lần cuối và phạm vi của mục tin tức được hiển thị
(4) mô tả liên quan đến kiểu thiết bị cuối của người dùng

Tất nhiên, nếu thông tin chi tiết hơn có thể dễ hiểu hơn thì không cần phải cố gắng không nhìn vào nó. Nếu có bất kì con số hoặc điều kiện nào thu hút được sự chú ý của bạn, hãy đi tới và gạch chân chúng. Nhưng tại thời điểm không cần thiết phải cố gắng để hiểu hoặc ghi nhớ chúng.

Tiếp theo, ta đọc [Mô tả tiến trình]. Đi theo luồng tiến trình trong biểu đồ đã cho, theo trình tự đọc. Chỉ cần biết rằng mô tả này giải thích từ (1) đến (4) cho mỗi tiến trình được chỉ ra trong biểu đồ và các nói rõ trong (5) là thông tin giữa các tiến trình chỉ được trao đổi khi sử dụng các tệp được chỉ ra trong biểu đồ.

[Câu hỏi con 1]

Hình đã cho chỉ ra nội dung của tệp người dùng, và câu hỏi con này hỏi về các giá trị của các nội dung nào đó (ngày bắt đầu trích rút và thời gian bắt đầu trích rút) của tệp tạm *A* đã được chuẩn bị bởi tiến trình nhập yêu cầu khi người dùng sử dụng hệ thống.

Đầu tiên, đọc bản ghi của người dùng. Sau đó đọc [Phác thảo hệ thống] (2), và biết rằng định danh người dùng là "XR205", người dùng đã đăng kí hai loại (“kinh tế” và “giải trí”) và lần cuối người dùng truy cập hệ thống này là 20:38 ngày 1 tháng 4 năm 2008. Tại thời điểm này, cần chú ý phần mô tả “hệ thống cung cấp tối đa 10 mục tin tức lùi dần theo ngày giờ cho mỗi loại được đăng kí”.

Điều được hỏi bao gồm các nội dung trong một tệp tạm được tạo bởi tiến trình nhập yêu cầu, nên tiếp theo bạn cần đọc (1) của [Mô tả tiến trình]. Ở đó, bạn sẽ thấy tệp tạm *A* bao gồm thông tin được yêu cầu (định danh người dùng, kiểu thiết bị cuối) và bản ghi đã lựa chọn (nội dung của tệp người dùng tương ứng với định danh người dùng), nhưng mô tả này không chỉ rõ ngày hoặc thời gian bắt đầu trích rút sẽ được sử dụng như thế nào. Vì thế bạn nên tiếp tục đọc tiếp. Khi bạn đọc đến (2), bạn sẽ thấy ngày giờ bắt đầu trích rút được sử dụng như những điều kiện trích rút bản ghi cho tệp tin tức. Nói cách

khác, chương trình so sánh ngày giờ đăng kí của bản ghi trong tệp tin tức với ngày giờ bắt đầu trích rút và trích rút những bản ghi đã đăng kí sau thời điểm bắt đầu trích rút ((2)(i)).

Ở đây, tin tức được trích rút từ tệp tin tức để chỉ hiển thị những mục tin tức mà thỏa mãn những điều kiện nào đó. Từ ý nghĩa của mục tin tức, ta thấy điều này liên quan đến điều kiện về thời gian. Nhớ rằng có một mô tả có phần liên quan đến điều này trong (3) của [12], vì thế cần đọc lại nó một lần nữa. Theo đó, các tình huống là khác nhau phụ thuộc vào việc liệu thời điểm khi người dùng truy cập vào hệ thống có qua 24 giờ sau thời điểm khi người dùng truy cập lần cuối vào hệ thống. Nếu bạn kiểm tra điều này cho người dùng thì kết quả là nhiều hơn 24 giờ, vì thế mọi mục tin tức đã đăng kí trong vòng 24 giờ cuối được đưa ra để trích rút. Do đó, ngày giờ bắt đầu trích rút trong tệp tạm A là những giá trị ám chỉ thời điểm 24 giờ trước thời điểm hiện tại.

Thời điểm hiện tại là khi người dùng sử dụng hệ thống này, đó là 17:00 ngày 3 tháng 4 năm 2008. Thời điểm 24 giờ trước đó là 17:00 ngày 2 tháng 4 năm 2008. Do đó ô trống A là (e) "20080402", và ô trống B là (b) "1700".

[Câu hỏi con 2]

Câu hỏi này về định dạng bản ghi trong tệp tạm *B*, đó là đầu ra của tiến trình trích rút tin tức. Vì thế bạn cần đọc lại (2) trong [Mô tả tiến trình]. Ở đó, bạn sẽ thấy tệp này dựa vào tệp tạm *A* và chứa các bản ghi được trích rút từ tệp tin tức cũng như vài thông tin cần thiết thêm vào, nhưng bạn sẽ không biết định dạng chi tiết của các bản ghi (cấu tạo mục).

Nói chung một tệp đầu vào là một tệp tổng kết mọi dữ liệu cần thiết cho tiến trình đó. Tệp tạm *B* được tạo ra để đưa tin tức ra trên thiết bị cuối của người dùng trong tiến trình đưa kết quả ra. Vì thế, cần cân nhắc xem liệu tất cả các mục cần thiết để tạo đầu ra có sẵn dùng hay không. Đây không phải là điều duy nhất cho câu hỏi này. Trong bất kì câu hỏi nào, nếu vị trí bạn điền vào ô trống trong một định dạng tệp thì cần chú ý tới nội dung đưa ra (tệp hay báo cáo in) sử dụng tệp đó và tìm kiếm các mục còn thiếu.

Có hai điều bạn phải cẩn thận trong điểm này. Một là trường hợp khi có một bước riêng khác giữa bước trong đó nội dung đầu ra liên quan đã sẵn sàng và bước đang xét. Trong trường hợp này, tiến trình của bước riêng giữa hai bước cũng phải được xem xét. Hai là liệu giá trị của mục tin đó có thể tìm thấy hay không trong bước tại đó tệp đưa ra để xem xét đang được tạo ra.

Giữa tiến trình trích rút tin tức nơi tệp tạm B được tạo ra và tiến trình đưa kết quả ra nơi nội dung đầu ra đã sẵn sàng, có một tiến trình sắp xếp tin tức. Nhưng vì bước này chỉ liên quan đến việc sắp xếp nên các tệp và giá trị tệp không thay đổi. Do đó, bạn không cần lo nghĩ về tiến trình sắp xếp tin tức, chỉ cần so sánh nội dung đầu ra của (4) với định dạng bản ghi của tệp tạm B. Hiển nhiên là định dạng bản ghi đang bị thiếu trường “loại”. Vì thế sau một chút băn khoăn, bạn cần nhớ lại mô tả của (5) trong [Mô tả tiến trình]. Vì thông tin trao đổi giữa các tiến trình là giới hạn với các tệp, bạn bắt đầu suy nghĩ, “Không có bất kì thông tin nào khác cần thiết để tạo nội dung đầu ra?” Nội dung đầu ra có nhiều định dạng phụ thuộc vào kiểu thiết bị cuối, vì thế bạn cần kiểu của thiết bị cuối. Lúc này, chúng ta có hai lựa chọn cho câu trả lời: “loại” và “kiểu thiết bị cuối”. Nếu bạn có thể tìm ra xem liệu những nội dung này có thể sẵn sàng trong tiến trình trích rút tin tức hay không, thì bạn có thể tìm ra câu trả lời đúng. Cả hai trường này được chứa trong tệp tạm A, tệp đầu vào, nên cả hai đều đúng . Vì thế, câu trả lời chính xác là (f) “loại” cho ô trống C và (a) “kiểu thiết bị cuối” cho ô trống D. Tất nhiên, thứ tự mà

chúng được viết là không liên quan, nhưng thứ tự được liệt kê ở đây là thích hợp trên phương diện của tệp tạm A và thành phần của nội dung đầu r
Không cần thiết phải chuyển đổi thứ tự.

[Câu hỏi con 3]

Câu hỏi này bao gồm những khóa sắp xếp cho tiến trình sắp xếp tin tức. Các câu hỏi về thiết kế của chương trình xử lý tệp hầu như luôn hỏi về khóa sắp xếp, vì thế bạn cần biết cách trả lời các câu hỏi này.

Không cần phải nói, khi đang xem xét các khóa sắp xếp, bạn cần hiểu được mục đích của việc sắp xếp. Thông thường, có bốn mục đích cho việc sắp xếp, vì thế cần biết những điều này và sử dụng chúng để tham khảo khi bạn xem xét các mục đích có thể để sắp xếp.

Bốn mục đích của sắp xếp

(i) **Đối sánh tệp**: để đối sánh giữa các tệp được tổ chức tuần tự, mỗi tệp phải được sắp xếp theo thứ tự giống với khóa đối sánh. Quá trình sắp xếp được thực hiện để chuẩn bị cho tiến trình này.
(ii) **Tính tổng**: khi tính tổng cho mỗi nhóm cụ thể, chẳng hạn như cho một số người dùng hoặc cho một ngày, dữ liệu trong cùng một nhóm cần đi cùng nhau. Sắp xếp được thực hiện bởi (các) mục khóa cho phân nhóm cụ thể .
(iii) **Xếp loại**: để xếp loại nhiều dữ liệu, dữ liệu phải được sắp xếp từ vị trí đầu tiên trở xuống theo mục được sử dụng để xếp loại. Sắp xếp được thực hiện nhằm mục đích này .
(iv) **Đưa ra các báo cáo**: Trong một báo cáo đưa ra, các nội dung cần thiết về cơ bản được in ra theo thứ tự của tệp đầu vào. Sắp xếp được thực hiện để điều khiển thứ tự của việc đưa dữ liệu ra trên báo cáo.

Trong tiến trình sắp xếp tin tức, tệp tạm B đã sẵn sàng trong tiến trình trích rút tin tức được sắp xếp đầu tiên, sau đó kết quả (tệp tạm C) được chuyển giao cho tiến trình đưa kết quả ra. Do đó, lý do cho việc sắp xếp là tiến trình đưa kết quả ra giả định rằng dữ liệu đã được sắp xếp. Nội dung của tiến trình đưa kết quả ra là trích rút tối đa 10 mục cho mỗi loại từ tệp tạm C, chỉnh sửa và ghi kết quả tùy theo kiểu thiết bị cuối. Vì giới hạn của quá trình trích rút

tối đa là 10 mục, bạn có thể nghĩ rằng mục đích không nằm trong bốn điều đã liệt kê ở trên. Tuy nhiên, những gì được trích rút không đơn thuần là lấy ra 10 mục một cách ngẫu nhiên mà trích rút theo quyền ưu tiên. Theo điểm (2) của [Phác thảo hệ thống], quyền ưu tiên là "có tối đa 10 mục tin tức lùi theo ngày giờ cho mỗi loại đã đăng kí", vì thế nó tương ứng với việc xếp loại. Với việc xếp loại này, các mục phải được sắp xếp cho mỗi loại từ những mục mới nhất theo ngày giờ xảy ra. Xếp loại từ những mục tin mới nhất theo ngày giờ xảy ra là sắp xếp từ lớn nhất (nghĩa là theo thứ tự giảm dần. Vì thế, câu trả lời cho ô trống E là loại (i). Đối với hai nhóm trả lời cuối, đáp án là (h) ngày xảy ra cho ô trống F và (g) thời gian xảy ra cho ô trống G..

[Câu hỏi con 4]

Điều được hỏi ở đây là số lượng bản ghi trong tệp tạm C và số mục tin tức hiển thị dưới các điều kiện đã cho. Vượt ra với câu hỏi, bạn đã gần như hiểu hầu hết nội dung tiến trình, vì thế bạn có thể trả lời mà không cần đọc câu hỏi lại. Nghĩa là, bạn có thể gần như dễ dàng theo kịp câu hỏi đến một mức nào đó mà chúng ta đã đọc lại phần giải thích kĩ càng hơn.

Từ bản ghi ta thấy người dùng đã đăng kí ba loại "chính trị", "kinh tế", "thể thao" và người dùng đã sử dụng hệ thống lần cuối vào lúc 4:00 ngày 1 tháng 4 năm 2008. Thời gian hiện tại của việc sử dụng là 11:00 ngày 1 tháng 4 năm 2008, vì thế nó nằm trong 24 giờ và phạm vi thời gian bắt đầu từ các bản ghi được trích rút từ 4:00 ngày 1 tháng 4 năm 2008.

Ở tệp tạm C, trong số tất cả các tin tức được trích rút trong suốt thời gian này, chỉ những tin tức trong ba loại trên được đưa ra. Số bản ghi này là, như được chỉ ra dưới đây, 50 (24 + 7 + 19). Do đó, ô trống H là (c).

Thời hạn cho việc trích rút →

Loại	Số lượng mục đã đăng kí trong m										
	00:00 ~ 00:59	01:00 ~ 01:59	02:00 ~ 02:59	03:00 ~ 03:59	04:00 ~ 04:59	05:00 ~ 05:59	06:00 ~ 06:59	07:00 ~ 07:59	08:00 ~ 08:59	09:00 ~ 09:59	10:00 ~ 10:59
Sức khỏe	0	0	0	0	0	0	2	0	1	0	0
Giáo dục	0	0	0	0	0	1	0	1	0	0	0

Tổng

Các loại đích cần trích rút

➡	Kinh tế	0	0	0	1	0	2	2	6	8	2	4	24
	Giải trí	1	0	1	0	1	2	0	1	2	1	1	1
	Khoa học	1	0	1	0	1	1	3	2	3	2	5	5
	Xã hội	0	1	2	0	1	0	2	3	5	4	3	3
➡	Thể thao	0	0	0	1	0	0	1	1	0	3	2	7
➡	Chính trị	0	1	0	0	0	1	4	2	3	6	3	19

Nói cách khác, với mỗi loại số lượng mục tin tức được hiển thị là tối đa là 10; vì thế số mục được hiển thị cho kinh tế và chính trị, đều vượt quá 10, sẽ là 10, và với thể thao, nhỏ hơn 10 mục, sẽ là 7. Vì ô trống I là tổng số mục được hiển thị, nên nó sẽ là (a) 27.

Câu hỏi 2

Q2. Đọc mô tả về thiết kết chương trình, sau đó trả lời các câu hỏi con từ 1 đến 3.

Một chương trình được phát triển cho việc truy vấn kiểm kê như là một phần của hệ thống quản lí hàng tồn kho. Chương trình truy vấn kiểm kê này nắm bắt 2 loại giao dịch.

Loại giao dịch "A" là các truy vấn về sự chuyển dịch (vào hoặc ra) của các thành phần trong một khoảng thời gian cụ thể. Một bảng các chuyển dịch thành phần là đầu ra của loại giao dịch này.

Loại giao dịch "S" là các truy vấn về hàng tồn kho hiện tại. Một bảng các trạng thái hàng tồn kho là đầu ra của loại giao dịch này. Nếu như có những phần trong hàng tồn nhỏ hơn giá trị nhỏ nhất của số lượng hàng tồn kho thì các phần này được thể hiện với màu đỏ.

Mỗi giao dịch có cấu trúc như hình bên dưới. Giá trị của các con số, loại dữ liệu,... trong mỗi trường được kiểm tra. Bên cạnh đó, hệ thống đòi hỏi một giá trị số nhằm xác nhận quyền sử dụng các bảng có quan hệ hàng tồn. Sự tồn tại của giá trị số xác nhận quyền sử dụng trong giao dịch cũng được kiểm tra trong bảng giá trị xác nhận quyền sử dụng. Các giao dịch không vượt qua được sự kiểm tra này sẽ được xử lí như các lỗi.

[Khuôn dạng giao dịch]

(1) Giao dịch loại "A"

A	Loại giao dịch	Thời điểm bắt đầu (năm, tháng, ngày)	Thời điểm kết thúc (năm, tháng, ngày)

Số lượng các số hiệu thành phần	Số hiệu thành phần	Số hiệu thành phần	. . .	Số hiệu thành phần

(2) Giao dịch loại "S"

A	Loại giao dịch

Số lượng các	Số hiệu	Số hiệu	. . .	Số hiệu

số hiệu thành phần	thành phần	thành phần		thành phần

[Thành phần trong cấu trúc dữ liệu có liên quan tới các truy vấn kiểm kê]

(1) Bảng thành phần

Số hiệu thành phần	Tên thành phần	Đơn vị sắp xếp	Số lượng hàng thấp nhất

(2) Bảng kiểm kê

Số hiệu thành phần	Số lượng hàng

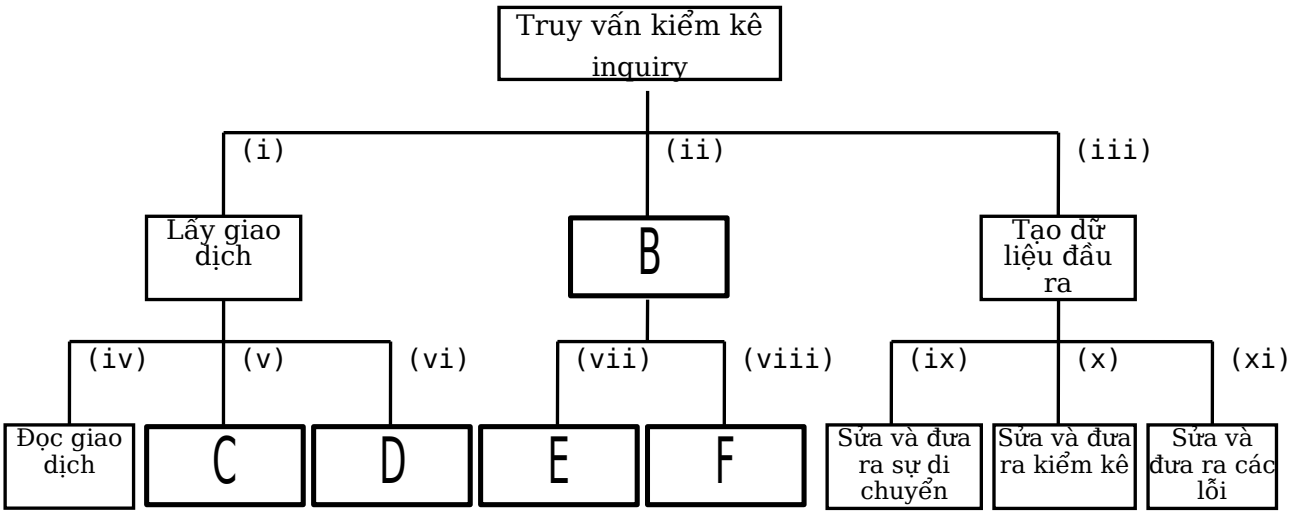
(3) Bảng giá trị xác nhận quyền sử dụng

Giá trị xác nhận quyền sử dụng

(4) Bảng di chuyển

Số hiệu thành phần	Vào/ra	Số lượng	Tem thời gian (năm, tháng, ngày, giờ, phút)

[Lược đồ cấu trúc mô đun]



[Các giao diện liên mô đun]

Số hiệu giao diện	Đầu vào	Đầu ra
(i)	–	Giao dịch, cờ kiểm ra khuôn dạng, cờ kiểm tra quyền sử dụng
(ii)	Giao dịch	Các bản ghi kết nối, cờ kiểm tồn kho, cờ thiếu số thứ tự thành phần
(iii)	Các bản ghi kết nối, G, cờ tồn kho thấp, cờ thiếu số hiệu thành phần, cờ kiểm tra khuôn dạng, cờ kiểm tra quyền sử dụng	–
(iv)	–	Giao dịch
(v)	Giá trị quyền sử dụng	Cờ kiểm tra quyền sử dụng
(vi)	Giao dịch	Cờ kiểm ra khuôn dạng
(vii)	Giao dịch	Các bản ghi kết nối (bảng các thành phần, bảng di chuyển), cờ thiếu số hiệu thành phần
(viii)	Giao dịch	Các bản ghi kết nối (bảng các thành phần, bảng tông kho), cờ tồn kho thấp, cờ số hiệu thành phần
(ix)	Các bản ghi kết nối (bảng các thành phần, bảng di chuyển)	–
(x)	Các bản ghi kết nối (bảng các thành phần, bảng tồn kho), cờ tồn kho thấp	–
(xi)	Giao dịch, H	–

[Cấu trúc cờ]

(1) Cờ kiểm ra khuôn dạng, cờ kiểm tra quyền sử dụng

Cờ

(Không lỗi: 0, Lỗi: 1)

(2) Cờ tồn kho thấp

Số các số hiệu thành phần	Cờ	Cờ	. . .	Cờ

(Trong thứ tự các số thứ tự thành phần trong giao dịch, lớn hơn hoặc bằng

giá trị hàng nhỏ nhất tồn tại: 0, nếu kiểm soát thấp hơn: 1)

(3) Cờ mất số thứ tự thành phần

Số các số hiệu thành phần	Cờ	Cờ	. . .	Cờ

(Theo thứ tự của các số hiệu thành phần trong giao dịch, số hiệu thành phần tồn tại: 0, số hiệu thành phần bị mất: 1)

Câu hỏi con 1

Từ nhóm các câu trả lời bên dưới, chọn ra câu trả lời đúng để điền vào các ô trống [ ] trong khuôn dạng giao dịch bên trên.

Nhóm câu trả lời cho A:

a) "A"
b) "AS"
c) "S"
d) Số các giao dịch
e) Di chuyển
f) Tem thời gian (năm, tháng, ngày, giờ, phút)
g) Cờ
h) Giá trị số xác nhận quyền sử dụng

Câu hỏi con 2

Từ nhóm các câu trả lời bên dưới, chọn ra câu trả lời đúng để điền vào các trống [ ] trong lược đồ cấu trúc đun bên trên.

Nhóm câu trả lời từ B tới F:

a) Lấy các bản ghi kết nối
b) Xử lí các bản ghi có liên quan tới các trạng thái hàng tồn.
c) Cập nhật bảng hàng tồn kho
d) Kiểm tra khuôn dạng giao dịch
e) Xử lí các bản ghi có liên quan tới các hoạt động di chuyển
f) Bảng nhập vào và phân phối
g) Tạo ra thông báo đặt hàng
h) Kiểm tra giá trị xác nhận quyền truy cập

Câu hỏi con 3

Từ nhóm các câu trả lời bên dưới, chọn ra câu trả lời đúng để điền vào các ô trống [ ] trong bảng các giao diện liên module ở trên.

Nhóm câu trả lời cho G:

a) Giao dịch
b) Loại giao dịch
c) Số hiệu giao dịch

d) Tem thời gian (năm, tháng, ngày, giờ, phút)
e) Số thứ tự thành phần
f) Số các số thứ tự thành phần
g) Tên thành phần
h) Giá trị xác nhận quyền sử dụng
i) Bảng giá trị xác nhận quyền sử dụng

Nhóm câu trả lời cho H :

a) Cờ kiểm soát thấp, cờ mất số thứ tự thành phần
b) Cờ kiểm soát thấp, cờ mất số thứ tự thành phần, cờ kiểm tra cấu trúc
c) Cờ kiểm soát thấp, cờ mất số thứ tự thành phần, cờ kiểm tra xác nhận quyền sử dụng
d) Cờ kiểm soát thấp, cờ kiểm tra cấu trúc
e) Cờ kiểm soát thấp, cờ kiểm tra cấu trúc, cờ kiểm tra xác nhận quyền sử dụng
f) Cờ kiểm soát thấp, cờ kiểm tra xác nhận quyền sử dụng
g) Cờ mất số thứ tự thành phần, cờ kiểm tra cấu trúc
h) Cờ mất số thứ tự thành phần, cờ kiểm tra cấu trúc, cờ kiểm tra xác nhận quyền sử dụng
i) Cờ mất số thứ tự thành phần, cờ kiểm tra xác nhận quyền sử dụng
j) Cờ kiểm tra cấu trúc, cờ kiểm tra xác nhận quyền sử dụng

Đáp án câu 2

Đáp án đúng:

[Câu hỏi con 1] A – h

[Câu hỏi con 2] B – a, C – h, D – d, E – e, F – b

[Câu hỏi con 3] G – a, H – h

Giải thích

Đây là một câu hỏi đề cập đến việc thiết kế chương trình tạo truy vấn trên dữ liệu kiểm kê. Các câu hỏi thành phần có liên quan đến dữ liệu giao dịch, biểu đồ cấu trúc module, và giao diện giữa các mô đun, tất cả trước đây thường xuất hiện dưới dạng các câu hỏi thiết kế nội dung chương trình trong các kỳ thi trước, do đó hãy chắc chắn rằng bạn hiểu cách trả lời những câu hỏi này.

[Câu hỏi con 1]

Ô trống được đặt vào khuôn dạng giao dịch và bạn được yêu cầu phải điền nội dung (tên khoản mục) vào các ô trống. Gợi ý là ô trống này được thiết lập cho các kiểu "A" và "S". Mặc dù chữ "giao dịch" có thể không quen thuộc với nhiều thí sinh, một cách cơ bản, nó có nghĩa là "dữ liệu vào". Chẳng hạn trong một chương trình trực tuyến như chương trình này, một "giao dịch" là nội dung được đưa vào trong màn hình nhập liệu. Ngoài ra, nhắc lại rằng phân chia TR, một kỹ thuật phân chia mô đun thường xuất hiện trong các kỳ kiểm tra, được sử dụng để phân chia các mô đun tiến trình ra từng kiểu khi có nhiều kiểu giao dịch.

Ta hãy quay lại với câu hỏi của chúng ta và đặt sự chú ý vào các phần có liên quan của văn bản câu hỏi dựa trên sự gợi ý đã đề cập ở trên. Vì mục này là chung với nhiều kiểu, bạn nên biết rằng hoặc là phải nhận ra kiểu giao dịch hoặc là phải chỉ ra một tiến trình chung cho các kiểu giao dịch. Để nhận ra được kiểu giao dịch, chúng ta đã xem xét một nhóm gọi là "kiểu dịch", do vậy đó có thể là đáp án đúng. Do đó, ta hãy chú tâm vào một tiến trình chung. Ngay trên [khuôn dạng giao dịch], đoạn văn bản chỉ ra rằng mỗi giao dịch "phải có một giá trị xác nhận quyền sử dụng để sử dụng được bảng quan hệ - hàng tồn kho. Sự tồn tại của giá trị xác nhận quyền sử dụng của giao dịch trong bảng các giá trị quyền sử dụng cũng được kiểm tra". Điều

này được thực hiện bất chấp kiểu giao dịch gì, và nó bị thiếu trong [khuôn dạng giao dịch]. Do đó, đáp án đúng là (h) giá trị xác nhận quyền sử dụng.

[Câu hỏi con 2]

Mục đích của câu hỏi này là điền vào ô trống trong biểu đồ cấu trúc mô đun. Nhóm câu trả lời chỉ ra tên mô đun, nhưng không có một sự giải thích chi tiết nào đề cập đến tiến trình,... do đó có thể có một số khó khăn. Trong m chừng mực nào đó, các tên mô đun có thể đem lại cho bạn một ý tưởng nào đó, nhưng trong các ô trống song song như C và D cũng như E và F, bạn có thể lo lắng về việc thứ tự sẽ được nhận biết như thế nào.**Tuy vậy, tất cả các câu hỏi đều được thiết kế theo cách chúng có thể trả lời được, phải có một gợi ý ở đâu đó.**Đừng lo lắng, hãy tìm câu trả lời. [Biểu đồ cấu trúc mô đun] có các số từ (i) đến (xi). Chú ý rằng những số này cũng được tìm thấy trong [Giao diện liên mô đun] như những con số giao diện. Do đó, chẳng hạn, mô đun tương ứng với ô trống C sẽ tương ứng với con số giao diện (v), chỉ ra rằng nó nhận một "giá trị xác nhận quyền sử dụng" từ mô đun lớp trên của nó là "lấy về giao dịch" và trả về một "cờ kiểm tra quyền sử dụng". Do đó, bạn biết được ô trống C phải là một mô đun thực thi một tiến trình liên quan đến việc kiểm tra các giá trị xác nhận quyền sử dụng. Tìm một câu trả lời đúng tương ứng trong nhóm các câu trả lời, bạn sẽ thấy câu trả lời đúng là (h) "Kiểm tra giá trị xác nhận quyền truy cập".

Bây giờ, bạn sẽ thấy cách để tìm ra những câu trả lời đúng, ta hãy tiến tới điền vào những ô trống còn lại. Mô đun ở ô trống D tương ứng với con số giao diện (vi), vì thế đầu vào là "giao dịch" và đầu ra là "cờ kiểm tra khuôn dạng". Do đó, đáp án đúng cho ô trống D là "kiểm tra khuôn dạng giao dịch". Các ô trống khác, B, E, và F có một quan hệ mật thiết. Thường thì trong những hoàn cảnh như thế này, mô đun - lớp trên B sẽ có tên bao trùm các mô đun ở dưới nó, chúng sẽ tương ứng với các ô trống E và F. Do đó chúng ta sẽ xem xét đến các ô trống E và F trước. Đối với ô trống E, con số giao diện là (vii), vì thế đầu vào là "giao dịch" và đầu ra là "các bản ghi kết nối (bảng các thành phần, bảng dịch chuyển), cờ mất số thứ tự thành phần". Từ thực tế là các bản ghi kết nối chứa bảng các thành phần cũng như bảng các dịch chuyển, bạn có thể dễ dàng hình dung ra được đó phải là một tiến trình tương ứng với các câu truy vấn liên quan đến các hành vi dịch chuyển của giao dịch loại "A". Tìm môt lựa chọn tương ứng với những điều này trong nhóm các câu trả lời, bạn sẽ tìm thấy (e) "Xử lí các bản ghi có liên quan tới các hoạt động di chuyển" và (f) "Bảng nhập vào và phân phối". Tuy nhiên, nhắc lại rằng mục đích của chương

trình là về các truy vấn trên dữ liệu hàng tồn kho. Vì nó là về các truy vấn, câu trả lời dường như không cần phải sửa đổi. Do đó, câu trả lời đúng là (e) "Xử lí các bản ghi có liên quan tới các hoạt động di chuyển" đối với ô trống E. Vì đầu ra cho ô trống F có chứa "các bản ghi kết nối (bảng các thành phần, bảng hàng tồn kho)" , bạn có thể thấy câu trả lời đúng phải là (b) "Xử lí các bản ghi có liên quan tới các trạng thái hàng tồn".

Bây giờ, ô trống còn lại là B. Đây là một mô đun thống nhất một tiến trình của các bản ghi liên quan đến các hành vi di chuyển và một tiến trình của các bản ghi liên quan đến trạng thái hàng tồn. Có một gợi ý khác. Ta đã sử dụng các câu trả lời (b), (d), (e) và (h) để điền vào các ô trống C, D, E và F. Văn bản câu hỏi không nói rằng các lựa chọn câu trả lời có thể được sử dụng nhiều lần, do đó ta có thể loại trừ các câu trả lời đã được lựa chọn. Ngoài ra, (c) cập nhật bảng hàng tồn còn (f) cập nhật bảng các dịch chuyển là các lựa chọn không thích hợp đối với một chương trình truy vấn kiểm kê, do đó chúng cũng bị loại trừ ra. Chúng ta bây giờ còn lại hai lựa chọn: (a): "lấy các bản ghi kết nối" và (g): "Tạo thông tin xếp đặt vị trí". Đối với câu hỏi truy vấn về trạng thái hàng tồn kho, văn bản câu hỏi chi ra rằng: "nếu có một thành phần nào đó trong hàng tồn kho có giá trị dưới số lượng dữ trữ nhỏ nhất thì như thành phần này được biểu thị màu đỏ". Bạn có thể dễ dàng hình dung ra rằng điều đó có nghĩa là "hãy chọn những thành phần này". Tuy nhiên, trong truy vấn của các hành vi dịch chuyển điều đó không thể xảy ra. Như đã đề cập trước đây, mô đun trong ô trống này hợp nhất hai mô đun tương ứng với hai ô trống E và F, vì thế (g) "Tạo thông báo đặt hàng" không phải là đáp án đúng. Trái lại, vì đầu ra của các giao diện của hai mô đun tương ứng với các ô trống E và F, nghĩa là trả về từ những mô đun này, đều chứa "các bản ghi kết nối", sẽ không có vấn đề gì nếu ta chọn đáp án (a) "lấy các bản ghi kết nối". Do đó, đây là câu trả lời đúng. Ngoài ra, mô đun bên trái của ô trống này là: "lấy giao dịch", do đó nội dung của các mô đun tương đương với nhau và đó là đáp án thích hợp.

[Câu hỏi con 3]

Cuối cùng chúng ta có những câu hỏi yêu cầu điền vào các ô trống trong [giao diện liên mô đun]. Nếu tiến trình của một mô đun thậm chí được mô tả một cách ngắn gọn, bạn có thể nghĩ tới nội dung cần thiết của đầu vào cũng như nội dung cần thiết của đầu ra khi xử lí kết quả, nhưng câu hỏi này không giống như thế. Tuy nhiên, hãy đừng nản chí. **Chắc chắn có những gợi ý ở đâu đó.** Tĩnh tâm và suy nghĩ cũng là một điều quan trọng. Cố gắng nhớ lại những loại mục nào cần thiết trong một giao diện liên mô đun. Những mục không cần thiết bao gồm những thứ chỉ được sử dụng trong một mô đun. Không phải những mục này hay là nội dung - có thể nhận được bằng việc tham chiếu tới tệp tin hoặc tính toán trong mô đun – cần nhận được từ mô đun lớp trên. Các mục cần phải xử lí qua các mô đun mức thấp được phân phát như các đầu từ cho các mô đun mức cao hơn, và các mục yêu cầu bởi các mô đun cấp cao hơn được trả về như các đầu ra từ các mô đun cấp thấp hơn. Như bạn thấy ở biểu đồ cấu trúc mô đun, các mô đun có cấu trúc phân cấp. Chẳng hạn, nội dung có tên “giao dịch” là cần thiết cho việc chỉnh sửa và đưa ra lỗi trong (xi). Và những gì nhận nội dung này được gọi là “đọc giao dịch” (iv). Tuy nhiên, giữa “đọc giao dịch” (iv) và “chỉnh sửa và đưa ra lỗi” (xi), không có một giao diện trực tiếp nào, vì thế thông tin không thể được phân phát trực tiếp. Điều này là hiển nhiên. Tuy nhiên, thường những loại điểm này được xem như một manh mối trong câu hỏi liên quan đến giao diện liên mô đun. Trong ví dụ này, ta hãy xem xét cách mà những nội dung giao dịch được phân phát từ (iv) tới (xi). Bạn sẽ thấy rằng đường dẫn là `(iv)` đọc giao dịch → `(ii)` nhận giao dịch → truy vấn trên dữ liệu hàng tồn kho → `(iii)` truy vấn dữ liệu đầu ra → `(xi)` chỉnh sửa và đưa ra thông báo lỗi. Nói cách khác, để tạo dữ liệu đầu ra (iii) – là mô đun cấp trên cho (ix), (x) và (xi), tất cả các mục được phân phát từ (ix) cho đến (xi) phải được thêm vào, trừ trường hợp có thể những mục đó nhận được từ chính trong mô đun và những mục đó nhận được từ mô đun cấp thấp hơn. Từ quan điểm này, ta hãy nhìn vào nội dung theo đó phải được phân phát từ (ix) đến (xi) (bao gồm tất cả các nội dung từ (ix) đến (xi)). Đó là “các bản ghi kết nối, cờ tồn kho thấp, giao dịch, và ô trống H”. Trái lại, hãy so sánh những nội dung đó với đầu vào của (iii), bạn sẽ thấy rằng (a) “giao dịch” bị thiếu. Do đó, đó là đáp án đúng cho ô trống G. Trong hệ thống thực, thường các giao dịch nhận lại được ở (iii) và (xi), dẫn đến khả năng bảo trì thấp. Tuy nhiên, trong một câu hỏi thi, bạn không cần phải lo lắng đến vấn đề này. Hãy nghĩ đơn giản và dễ hiểu.

Bây giờ, đối với ô trống H, ta hãy nghĩ ngược lại. Tất cả các mục mà (iii)

nhận được từ mô đun cấp trên của nó, chẳng hạn truy vấn trên dữ liệu tồn kho, hoặc là sẽ được sử dụng bởi chính (iii) hoặc sẽ được phân phát cho mô đun cấp thấp hơn (hoặc cả hai, tất nhiên). Vì quá trình của mỗi mô đun không được nói rõ trong câu hỏi này, hiện tại, chúng ta hãy giả sử rằng tất cả được giao cho các mô đun cấp thấp. Khi đó bạn sẽ thấy rằng, "cờ thiếu số hiệu thành phần, cờ kiểm tra khuôn dạng, và cờ kiểm tra xác nhận quy dụng" được sử dụng. Có một cờ khác – Cờ tồn kho thấp, nhưng nó đã được (x) dùng. Trong câu truy vấn trạng thái tồn kho, nếu lượng hàng tồn của một thành phần nào đó là thấp (dưới một số lượng dự trữ tối thiểu nào đó), thành phần đó được biểu thị màu đỏ, do đó, lượng hàng tồn kho thấp rõ ràng không được xử lí như một lỗi. Mặt khác, bạn cũng có thể hiểu rằng, ba cờ khác chỉ ra rằng một lỗi nào đó trong nội dung của giao dịch, đòi hỏi phải có m thông điệp báo lỗi được chỉnh sửa và trả về. Do đó, câu trả lời đúng là (h) "cờ thiếu số hiệu thành phần, cờ kiểm tra khuôn dạng, cờ kiểm tra quyền dụng". Thực tế, có thể sẽ không có vấn đề gì ngay cả khi cờ tồn kho thấp được chứa ở đây, nhưng không có một lựa chọn nào trong nhóm các câu trả lời chứa bốn cờ trên.

Trong một bài kiểm tra, sẽ có các câu hỏi liên quan đến thiết kế chương trình và thiết kế trong; tuy nhiên, không phải là "thiết kế" theo đúng nghĩa của từ đó. **Văn bản câu hỏi chứa một lí do cho mỗi câu trả lời đúng do đó, chỉ có một câu trả lời đúng khả dĩ.** Đồng thời cũng chỉ có một vài mẫu các câu hỏi. Do đó, hãy học các câu hỏi đã thi để học được phương phá (cách tìm ra lí do cho những câu trả lời), và khi đó bạn có thể chắc chắn tìm thấy những câu trả lời đúng.